



# How to Get Started with IBM WebSphere Application Server on Solaris 10 and Zones

Dileep Kumar  
Staff Engineer  
Market Development Engineering  
Sun Microsystems, Inc.

Albert Leigh  
Solution Architect  
Client Solutions  
Sun Microsystems, Inc.

**May 24, 2006**

# Table of Contents

How can I get started with WebSphere Application Server on Solaris 10?.....	3
Solaris 10 Containers and Zones.....	3
Two Types of Zones.....	4
Considerations.....	5
Step-by-Step Guide.....	5
Overview of the Procedure.....	7
I. Prepare the Solaris 10 Environment for WebSphere Application Server V6.....	7
Changes in the Solaris 10 System Tunable Parameters.....	9
TCP Tunings.....	11
II. Installing WAS V6 in the Global Zone.....	11
III. Create 3 non-global zones that inherit the WAS installation from the global zone and configure WebSphere.....	13
IV. Create another non-global zone and install Sun Java System Web Server..	19
V. Install WebSphere Plug-in on Sun JS Web Server.....	20
VI. Remove Unwanted Zones.....	21
Conclusions and Summary.....	21
Additional Resources.....	23
Other References.....	23

## How can I get started with WebSphere Application Server on Solaris 10?

This document describes how **IBM WebSphere Application Server (WAS) V6.0.2** or newer can be installed and configured on the Sun Solaris 10 Operating System in the global and non-global zones. Since product documentation already exists for WAS planning, installation and deployment from IBM, we do not repeat such information and only the important step-by-step guidelines for Solaris 10 and Zones are summarized here. We list and provide links to the existing documentation in the “Additional Resources” section.

Consider this document as a Quick Start guide for WAS Deployment on Solaris 10. It can also supplement IBM WebSphere documentation such as *Getting Started Guide* and *Installation Guide*. <http://www-306.ibm.com/software/webservers/appserv/was/library>

For the recommended fixes for WAS, please refer to this IBM site:

<http://www-1.ibm.com/support/docview.wss?rs=180&uid=swg27004980>

**Note:** IBM SW supports WAS 5.1.x only in global zone on Solaris 10.

### **Solaris 10 Containers and Zones**

The Solaris 10 Operating System provides many advanced features and is an excellent choice for enterprise application deployment. J2EE-based enterprise applications running on previous versions of Solaris can be deployed on Solaris 10 without any changes and may take advantage of Solaris Container and Zone technology.

Solaris Containers can be built using one or more the following technologies. These technologies can be combined to create Containers tailored for a specific server consolidation project.

- Solaris Resource Manager, for workload resource management
- Resource Pools, for partitioning
- Zones, for isolation, security and virtualization

It is important to note the terminology of Solaris Container and Zone. Solaris Containers are a combination of Resource Management, System Administration (customization) and Zones. Zones is one of the technology that enables Containers. Zones technology can be used to create a Container with certain characteristics, such as the isolation provided by the virtual Solaris environment. But it is also possible to create another Solaris Container, without Zone, using Resource Pools technology if the required characteristics of that Container can be met with the features Resource Pools provide. So while a Zone is a Container, a Container is not necessarily a Zone.

**Note:** Solaris Resource Manager and Resource Pools technologies existed before Solaris 10.

Solaris Zones provide the following features:

- Security—Network services can be run in a zone, limiting the damage that can be done to the system and other zones in the event of a security violation.
- Isolation—Applications requiring exclusive access to global resources, such as specific usernames or network ports, can run on the same machine using Solaris Zones. Each zone has its own namespace, completely separate from other zones. Users in a zone are unable to monitor other zones, such as viewing network traffic or the activity of processes.
- Virtualization—Solaris Zones present a virtualized environment to applications, removing the physical details of the hardware from view. This eases redeployment of applications on a different physical machine.
- Granularity—Since Solaris Zones are implemented in software, zones are not limited

to the granularity defined by hardware boundaries. Instead, zones offer sub-CPU granularity. Zones do not require dedicated CPU resources, dedicated I/O devices such as host bus adapters and network interface cards, or dedicated physical memory. As a result, even a system with a single processor can be used to host several zones.

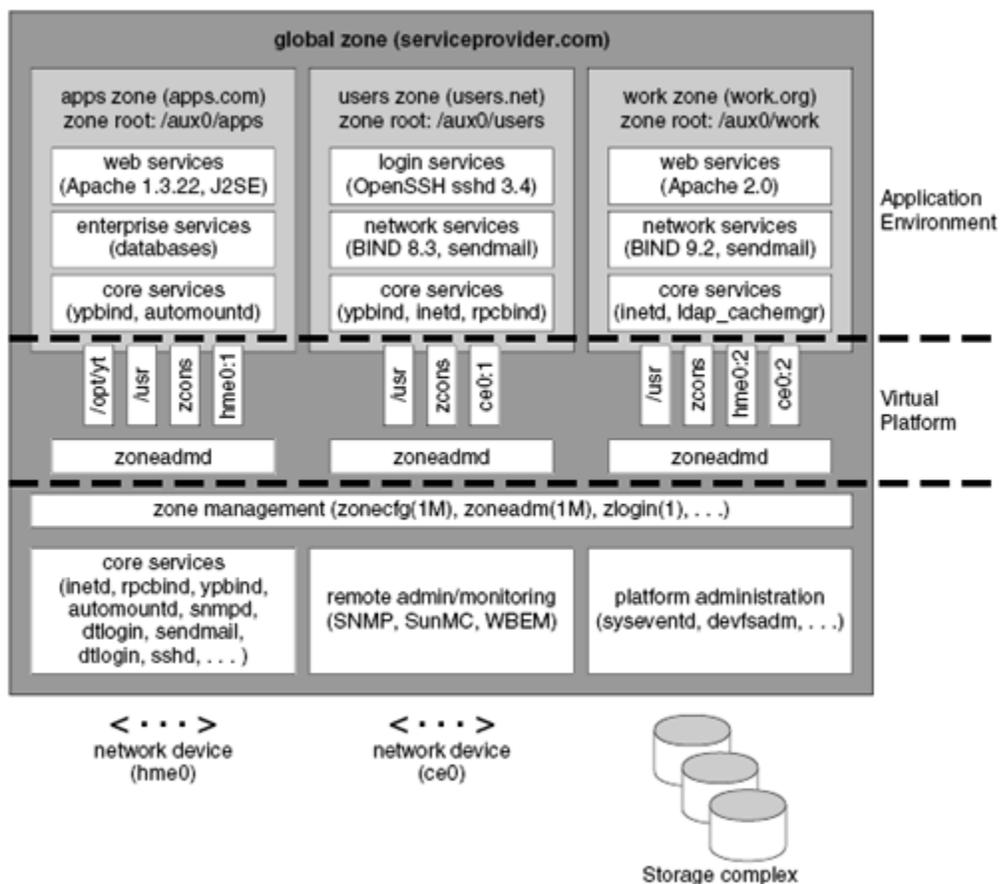
- **Transparency**—The environment presented to the application in a zone is nearly identical to the standard Solaris OS environment. There are no new, zone-specific application programming interfaces (APIs) or application binary interfaces (ABIs) to which applications must be ported. Some restrictions do exist due to security and isolation requirements. These restrictions mainly affect applications that perform privileged operations or need access to physical devices.

## Two Types of Zones

Two types of zones are available—the global zone, and non-global zones.

The global zone encompasses the entire system. Because it is equivalent to a typical Solaris OS instance, the global zone has access to the physical hardware, and can control all processes.

Non-global zones are located inside the global zone, and are isolated from the physical characteristics of the system. They are also referred to as local zones (unofficially).



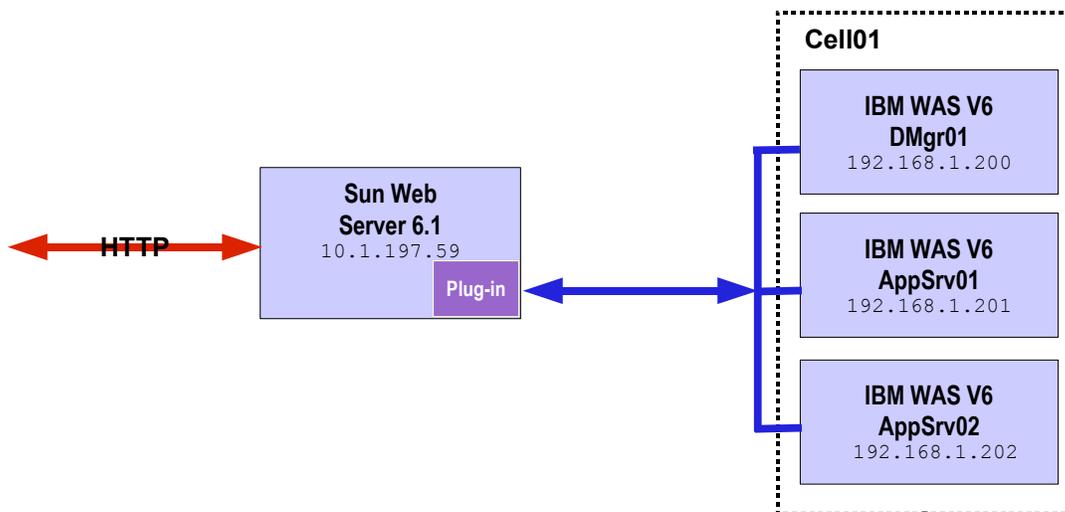
**Figure 1: Solaris Zones**

We will discuss further details on how you can install and configure the IBM WebSphere Application Server using Solaris 10 and Zone technologies.

## **Considerations**

Multi-Tier web applications are a prime driver of server sprawls in the data center generated by the typical affinity between applications and dedicated physical servers. Web and Application Servers, like IBM WebSphere, comprise a large majority of these installs. The server sprawls drive up operational costs due to real estate, air conditioning, and energy consumption. Server consolidation can reduce such operational costs while improving utilization, security, manageability and efficiency in data center operations. It can be accomplished with Solaris Container technologies and a new generation of high performance and power efficient Sun's server products.

Solaris Zones can help accomplish a simple consolidation effort. To demonstrate it, assume the following existing environment where application services are deployed on multiple physical servers (Figure 2).



**Figure 2: A WebSphere Application Server Environment without Solaris Zone**

In this environment, the Web Server is configured on the public network, while the WebSphere application environment is on a separate private network for security. The communication between the Web Server and the WebSphere environment is achieved by the Web Server plug-in. In the next section, we provide the basic steps needed to consolidate these services to one physical server using Solaris Zone. Additional topics on Solaris Containers, Resource Control, and Workload Management can be found in the references section.

## **Step-by-Step Guide**

The system used in this test environment is as follows:

- Sun Fire v490
  - 4x1.05GHz UltraSparc IV (Dual-Core) CPU's
  - 16 GB RAM
  - 2 Network Interface (ce0 and ce1)
  - Solaris 10 03/05

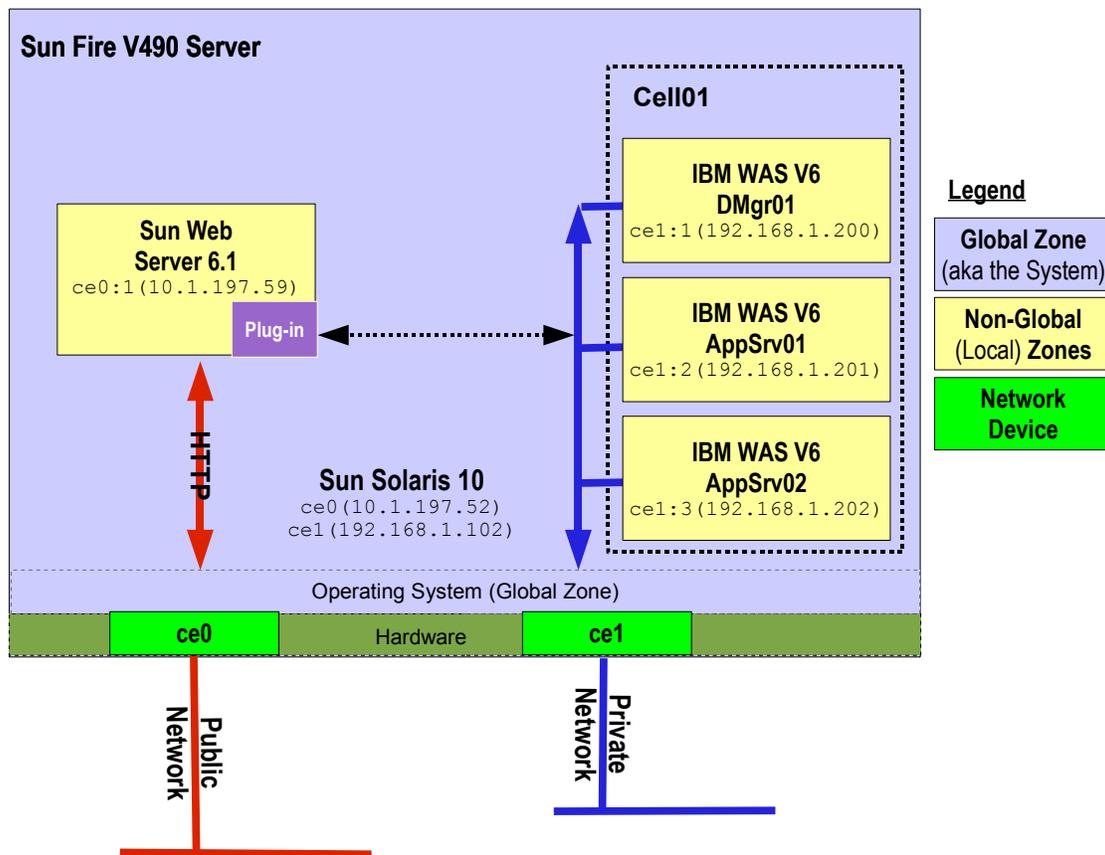
- IBM WebSphere Application Server Network Deployment (ND) V6.0.2 for Solaris/SPARC
- IBM WebSphere Application Server V5.1 for Solaris/SPARC
- Sun Java System Web Server V6.1 SP2

As shown in Figure 3, the Sun Fire v490 system is configured with Solaris 10 OS. The two network devices have been plumbed on two separate networks as follows: ce0 on 10.1.197.x and ce1 on 192.168.1.x. By default, Solaris 10 includes a global zone. Additional zones, also known as non-global or local zones, can be created to partition the single instance of Solaris installation into multiple virtual OS instances. The figure depicts five non-global zones that are also plumbed to logical network interfaces on ce0 and ce1:

1. Sun Java System Web Server on ce0:1
2. IBM WAS V6 Deployment Manager (Dmgr01) on ce1:1
3. IBM WAS V6 node (AppSrv01) on ce1:2
4. IBM WAS V6 node (AppSrv02) on ce1:3

The logical network devices assignment to zones, such as ce1:1 and ce1:2 is for an example only. They vary on device types (e.g. ce0, hme0, etc.) and order of creation (e.g. Zone 1 may get ce1:3 while Zone 2 may get ce1:1).

The server is pre-installed with Solaris 10 OS and network configurations. The following steps will guide you through to successfully complete the WebSphere Application Server environment configuration and installation as depicted in Figure 3.



**Figure 3: A WebSphere Application Server Environment with Solaris Zone**

## Overview of the Procedure

The following sections will guide you through the steps of installing WebSphere on Solaris 10.

- I. Prepare the Solaris 10 environment for WebSphere Application Server V6
- II. Install WAS V6 in the Global Zone. Do not create any profiles.  
IP Address: 192.168.1.102
- III. Create 3 non-global zones that inherit the WAS installation (/opt/IBM/WebSphere) from the global zone.
  1. Create a Deployment Manager Profile on waszone0. IP: 192.168.1.200
  2. Create a new profile in each zone (AppSrvZone1 on waszone1 and AppSrvZone2 on waszone2). IP: 192.168.1.201 and IP: 192.168.1.202
- IV. Create another non-global Zone to install Sun Java System Web Server. IP: 192.168.1.203
- V. Install WebSphere Plug-in on Sun JS Web Server 192.168.1.203.
- VI. Remove unwanted zones.

### I. Prepare the Solaris 10 Environment for WebSphere Application Server V6

The following is an IBM recommended kernel parameter values in the /etc/system file for Solaris.

```
set shmsys:shminfo_shmmax = 4294967295
set shmsys:shminfo_shmseg = 1024
set shmsys:shminfo_shmmni = 1024
set semsys:seminfo_semaem = 16384
set semsys:seminfo_semni = 1024
set semsys:seminfo_semmap = 1026
set semsys:seminfo_semns = 16384
set semsys:seminfo_semns1 = 100
set semsys:seminfo_semopm = 100
set semsys:seminfo_semnu = 2048
set semsys:seminfo_semume = 256
set msgsys:msginfo_msgmap = 1026
set msgsys:msginfo_msgmax = 65535
set rlim_fd_cur=1024
```

In the Solaris 10 release, many kernel tunable parameters including System V IPC facilities are either automatically configured or can be controlled by resource controls. Facilities that can be shared are memory, message queues, and semaphores.

Resource controls allow IPC settings to be made on a per-project (i.e. accounting group) or per-user basis on the local system or in a name service environment. In previous Solaris releases, IPC facilities were controlled by kernel tunable parameters. You had to modify the /etc/system file and reboot the system to change the default values for these facilities.

Because the IPC facilities are now controlled by resource controls, their configurations can be modified while the system is running. Many applications that previously required system tuning to function might now run without tuning because of increased default values and the automatic allocation of resources.

Most default kernel parameters are typically sufficient and good starting point to support WebSphere Application Server. In Solaris 10, many kernel values have been increased to accommodate applications' increased demand for more system resources. For installation and verification tests, we verified the values of these tunable parameters (installed as "root" user) and achieved successful installation and execution of WebSphere Application Server.

On Solaris 8 or 9, the following command was used to provide the IPC Module related setting:

```
bash-# sysdef -i
```

But, when this command is executed on Solaris 10 system, the output shows that these modules do not have system-wide limits. Below is the portion of output from “`sysdef -i`” command on Solaris 10.

```
*
* Process Resource Limit Tunables (Current:Maximum)
*
0x00000000000000100:0x0000000000010000   file descriptors
*
* Streams Tunables
*
    9 maximum number of pushes allowed (NSTRPUSH)
 65536 maximum stream message size (STRMSGSZ)
   1024 max size of ctl part of message (STRCTLSZ)
*
* IPC Messages
*
* The IPC Messages module no longer has system-wide limits.
* Please see the "Solaris Tunable Parameters Reference Manual" for
* information on how the old limits map to resource controls and
* the prctl(1) and getrctl(2) manual pages for information on
* observing the new limits.
*
*
* IPC Semaphores
*
* The IPC Semaphores module no longer has system-wide limits.
* Please see the "Solaris Tunable Parameters Reference Manual" for
* information on how the old limits map to resource controls and
* the prctl(1) and getrctl(2) manual pages for information on
* observing the new limits.
*
*
* IPC Shared Memory
*
* The IPC Shared Memory module no longer has system-wide limits.
* Please see the "Solaris Tunable Parameters Reference Manual" for
* information on how the old limits map to resource controls and
* the prctl(1) and getrctl(2) manual pages for information on
* observing the new limits.
*
```

To obtain the IPC and other settings for the current shell environment where the WebSphere Application Server is to be installed:

```
bash-3.00# prctl $$
```

This will list all the System V related IPC and file descriptor setting for the current shell and which will be applied to any process started within that shell. To make any changes, change the settings, close the current shell and log back in to get the new settings into effect.

**Note:** You must install WebSphere Application Server as `root` user. One reason is the WebSphere V6 installer invokes `pkgadd` that needs write privileges in system directories. Therefore, all the setting can be applied and obtained with the existing project called “`user.root`”. If you prefer, you may create a customer project, for instance “`websphere`”, and assign the root user to be member of that project.

For simplicity, it is assumed here that the “`user.root`” project is used instead of creating the new project. Do the following to get the current project id of the root user.

```
bash-# id -p
uid=0(root) gid=0(root) projid=1(user.root)
```

## Changes in the Solaris 10 System Tunable Parameters

Table 1 below lists the tunable kernel parameters, such as SYS V IPC, requiring adjustment for WebSphere Application Server installation on Solaris 8/9. It then provides the mapping from IBM's recommendation of SYS IPC parameters for WebSphere on Solaris 8/9 (*/etc/system*) to Solaris 10 (*/etc/project*). It also lists the new default values and obsoleted parameters.

IBM Recommended <i>/etc/system</i> Settings for WebSphere App Server	New Resource Control Parameters in Solaris 10	New Default Value
set shmsys:shminfo_shmmax = 4294967295	project.max-shm-memory	1/4 of physical memory
set shmsys:shminfo_shmseg = 1024	Obsoleted	
set shmsys:shminfo_shmmni = 1024	project.max-shm-ids	128
set semsys:seminfo_semaem = 16384	Obsoleted	
set semsys:seminfo_semmni = 1024	project.max-sem-ids	128
set semsys:seminfo_semmap = 1026	Obsoleted	
set semsys:seminfo_semmns = 16384	Obsoleted	
set semsys:seminfo_semmsl = 100	process.max-sem-nsems	512
set semsys:seminfo_semopm = 100	process.max-sem-ops	512
set semsys:seminfo_semnu = 2048	Obsoleted	
set semsys:seminfo_semume = 256	Obsoleted	
set msgsys:msginfo_msgmap = 1026	Obsoleted	
set msgsys:msginfo_msgmax = 65535	Obsoleted	
set rlim_fd_cur=1024	process.max-file-descriptor	256

**Table 1: Solaris 10 Tunable Parameters concerning WebSphere**

As we go through the IBM recommended */etc/system* settings in Solaris 10, we will ignore the obsoleted parameters; thus, we discuss only the following parameters:

- ▶ project.max-shm-memory
- ▶ project.max-shm-ids
- ▶ project.max-sem-ids
- ▶ process.max-sem-nsems
- ▶ process.max-sem-ops
- ▶ process.max-file-descriptor

If a parameter's current value is less than the recommended threshold, we will update it with that value. By using the *projmod* command, the settings are stored in the */etc/project* file.

```
bash-3.00# projmod -s -K 'project.max-shm-memory=(privileged,4gb,deny)' user.root
bash-3.00# projmod -s -K 'project.max-shm-ids=(privileged,1024,deny)' user.root
bash-3.00# projmod -s -K 'project.max-sem-ids=(privileged,1024,deny)' user.root
bash-3.00# projmod -s -K 'project.max-sem-nsems=(privileged,512,deny)' user.root
bash-3.00# projmod -s -K 'project.max-sem-ops=(privileged,512,deny)' user.root
bash-3.00# projmod -s -K 'project.max-file-descriptor=(privileged,1024,deny)' \
user.root
```

We can now examine the new tunable parameter settings. These project settings should be done to each zone where WebSphere is being installed.

```
bash-3.00# cat /etc/project
system:0::::
user.root:1::::
    process.max-file-descriptor=(privileged,1024,deny);
    process.max-sem-ops=(privileged,512,deny);
    process.max-sem-nsems=(privileged,512,deny);
    project.max-sem-ids=(privileged,1024,deny);
    project.max-shm-ids=(privileged,1024,deny);
    project.max-shm-memory=(privileged,4294967296,deny)
noproject:2::::
default:3::::
group.staff:10::::
```

We can also make these changes using the `prctl` command, but the settings will not persist.

```
bash-3.00# prctl -n project.max-shm-memory -r -v 4gb -i project 1
```

After making any changes to make the change effective we must logout and then login to take these changes into effect, or simply we can start a new shell window which will have the new settings in affect. Use `prctl` to examine the new settings.

```
bash-3.00# prctl $$
```

**Note:** We will assume that in every zone where we install WebSphere, these system parameters will be set accordingly. We will also have to adjust these parameters as part of the tuning process.

In Solaris 10, using `rctladm` we can enable logging so the system will notify us when we are running out of these resources. For example, If a user wants the system to notify when it is running out of “`process.max-file-descriptor`”, then issue the following command from the shell prompt:

```
bash-3.00# rctladm -e syslog process.max-file-descriptor
```

Modify `process.max-file-descriptor` in `/etc/rctladm.conf` file.

**Before:** `process.max-file-descriptor=none`

**After:** `process.max-file-descriptor=syslog=notice`

If the system happens to run out of file-descriptors, it will be reported in `/var/adm/messages` file:

```
Sep 27 13:57:04 isv-ibm-02 genunix: [ID 883052 kern.notice] privileged rctl
process.max-file-descriptor (value 5) exceeded by process 16797
Sep 30 17:43:19 isv-ibm-02 genunix: [ID 883052 kern.notice] basic rctl
process.max-file-descriptor (value 256) exceeded by process 10535
```

This can be achieved by directly updating the `/etc/rctladm.conf` file also. To get notification about all the resources, we can change all the line to as above for file descriptor and we can watch the `/var/adm/messages` file for notifications.

Set or get limitations on the system resources available to the current shell and its descendents (e.g. file-descriptors limits):

```
bash-3.00# ulimit
```

To get the values:

```
bash-3.00# ulimit -a
```

To set the values:

```
bash-3.00# ulimit -n <new value>
```

## TCP Tunings

TCP driver parameters can be set as follows:

- ▶ `ndd -set /dev/tcp tcp_conn_req_max_q 8192`
- ▶ `ndd -set /dev/tcp tcp_conn_req_max_q0 8192`
- ▶ `ndd -set /dev/tcp tcp_max_buf 4194304`
- ▶ `ndd -set /dev/tcp tcp_cwnd_max 2097152`
- ▶ `ndd -set /dev/tcp tcp_recv_hiwat 400000`
- ▶ `ndd -set /dev/tcp tcp_xmit_hiwat 400000`

## II. Installing WAS V6 in the Global Zone

The following steps demonstrate the basic steps to complete the WebSphere Application Server Network Deployment (WAS ND) installation on Solaris. In this scenario, the WebSphere installation directory in the global zone is shared by multiple WebSphere profiles in non-global zones including the Deployment Manager and Application Server nodes. Such sharing can save each WAS installation about 900MB in disk space. The product binaries (core product files) are installed in the global zone, but WAS is not configured. The configuration of each WAS profile is done in a respective non-global zone. Of course, one or more profiles of WAS can be configured in the global zone in this set up, as well.

Upon completion of installation, a few additional steps need to be performed in order for the core installation to be shared in multiple non-global zones. That includes modification of the WebSphere Profile properties.

Set `mozilla` to be the default browser.

```
bash-3.00# export BROWSER=/usr/sfw/bin/mozilla
```

**Note:** For readability in this document, we change the prompt to "global#" in this Bash shell. The non-global zones' prompts will have their own distinctive host names.

```
bash-3.00# export PS1="global\\$ "  
global#
```

If needed, set the X-Windows display to the appropriate workstation display id and set the shell's search path. Also, verify the `umask` to ensure proper access privileges.

```
global# export DISPLAY=${WORKSTATION}:1.0  
global# export PATH=$PATH:/usr/dt/bin:/usr/openwin/bin:/usr/sfw/bin  
global# umask 0022  
0022
```

Go to the WebSphere software distribution CD or directory and start up `launchpad.sh`. (**Note:** If you wish to do a batch install, modify the `responsefile.txt` and run with the command `install -options responsefile.txt`). In this example, WAS will be installed in the default `/opt/IBM/WebSphere` directory. If you wish to create a profile in the global zone, you can run `firststeps.sh` or the profile creation utility (e.g. `pctSolaris.bin`) at this point. If not, delay this step until you complete creating zones.

```
global# cd ${WAS_SW_DIST}
global# ./launchpad.sh
```

In WAS V6, IBM introduced profiles. A profile defines a run time execution environment and it has its own directory structure to store configuration files, the default location for deployed applications, logs, and other data. We must create profiles as the last step of WebSphere installation. We will demonstrate this step in non-global zones.

By default, profile creation will write into the directories like `/opt/IBM/WebSphere/logs` and `/opt/IBM/WebSphere/AppServer/profiles`. In order to share a single WebSphere installation in multiple zones, we will need to make the profile data to be written to locations other than the installation's root. To achieve this, we modify the WebSphere profile property file to point to a local path to each zone.

Go to the WebSphere properties directory.

```
global# cd /opt/IBM/WebSphere/AppServer/properties
```

Back up first and edit the file `wasprofile.properties`.

```
global# cp -p wasprofile.properties wasprofile.properties.orig
global# vi wasprofile.properties
# 5724-I63, 5724-H88 (C) COPYRIGHT International Business Machines Corp. 2000,
2004
# All Rights Reserved * Licensed Materials - Property of IBM

#-----
# The log home property determines the directory that would hold log
# files produced by the wasprofile tool.
#
# The default path is <install location>/logs/wasprofile.
#-----
WS_CMT_LOG_HOME=${was.install.root}/logs/wasprofile

#-----
# The prefix for all wasprofile log file names.
#-----
WS_WSPROFILE_LOG_NAME_PREFIX=wasprofile
#-----
# The profile registry property determines the path to the XML file that
# contains information about all registered profiles.
#
# The default path for this file is:
# <install location>/properties/profileRegistry.xml
#-----
WS_PROFILE_REGISTRY=/opt/WASProfiles/properties/profileRegistry.xml
.....
~
~
"wasprofile.properties" 32 lines, 1531 characters
```

Change the following two entries in the file `wasprofile.properties`:

**Before:** `WS_CMT_LOG_HOME=${was.install.root}/logs/wasprofile`  
**After:** `WS_CMT_LOG_HOME=/opt/WASProfiles/logs/wasprofile` and

**Before:** `WS_PROFILE_REGISTRY=${was.install.root}/properties/profileRegistry.xml`  
**After:** `WS_PROFILE_REGISTRY=/opt/WASProfiles/properties/profileRegistry.xml`

We will create `/opt/WASProfiles` in each non-global zone where WebSphere will be configured.

Examine /etc/hosts (In this example, we are using files).

```
global# cat /etc/hosts
#
# Internet host table
#
127.0.0.1        localhost
10.1.197.52     system52        loghost
10.1.197.59     system59        webserv1
#
# Private Net for IBM WebSphere Software Test
#
192.168.1.102   websvcs1
192.168.1.103   db2svr1
192.168.1.200   wasdmzone
192.168.1.201   waszone1
192.168.1.202   waszone2
192.168.1.205   was_v5
```

Examine the network interfaces.

```
global# ifconfig -a
```

In this example, we also need to set up a network route so that the Web Server on 10.1.197.59 can reach WAS nodes on 192.168.1.x network.

```
global# route add default 192.168.1.1
```

This route information will not last through a zone re-boot, so we will need to define this in either Solaris 10 Service Management Facility (SMF) or the old fashioned way with /etc/rc3.d.

*Note: Take caution in setting up network route as this can compromise security in your environment. If you need to block network traffic between non-global zones, use the "route reject" command.*

We now complete installing the WebSphere Application Server binaries. We proceed to configure the Solaris Zones.

### III. Create 3 non-global zones that inherit the WAS installation from the global zone and configure WebSphere

Create a script to configure the first WebSphere zone. This is a minimum requirement to create a zone.

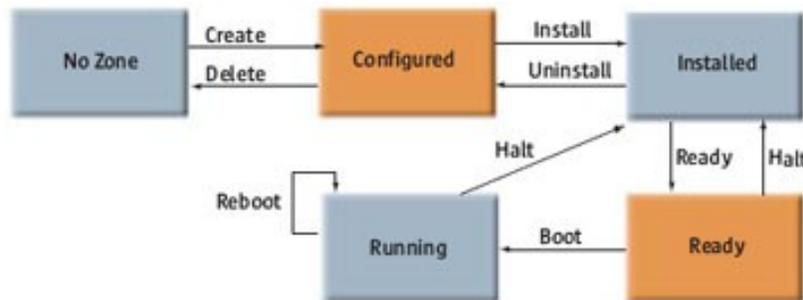
```
global# vi /waszone/wasdmzone.cfg
#
# Script to create WebSphere Deployment Manager Zone
# Author: Albert Leigh, Nov 18, 2005
#-----
# Create the zone
create
# zonepath is where the zone files reside
set zonepath=/export/zones/wasdmzone
# boot this zone when the system comes up
set autoboot=true
# configure the network interface
add net
# bind to ce1 device. It'll get a logical device like ce1:1
set physical=ce1
# and assign IP address. /24 is the netmask (24x1's)
set address=192.168.1.200/24
end
```

```

# make the global WAS installation available in this zone
# this is also known as LOP Back File System (lofs)
# and it is mounted read-only in this zone
add inherit-pkg-dir
    set dir=/opt/IBM/WebSphere
end
# ensure validity of this script
verify
# Just do it
commit
#
# End of Script
#

```

Use the script to create the zone. Figure 4 shows the life cycle of a zone.



**Figure 4: The Life Cycle of a Zone**

Display all configured zones. No zone means only the global zone is in existence.

```

global# zoneadm list -cv
global

```

Configure the first non-global zone using the script from Step#1.

```

global# zonecfg -z wasdmzone -f /waszone/wasdmzone.cfg

```

Install the zone.

```

global# zoneadm -z wasdmzone install
Preparing to install zone <wasdmzone>.
Creating list of files to copy from the global zone.
Copying <2579> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <1032> packages on the zone.

```

Boot the zone and log in to the console to configure the zone with necessary information such as language, locale, timezone, hostname, name service, root password, etc. just like doing a “Sys Config” for a fresh Solaris environment.

```

global# zoneadm -z wasdmzone boot
global# zlogin -C wasdmzone

```

(Sample Input to set up the new zone

```

Language: 0. English
Locale: 0. English (C - 7-bit ASCII)
Terminal: 12. Xterm
Hostname: xxxxxx (press F2 or ESC-2)
Kerberos Security: No
Name Service: None

```

```

Continent and Oceans: Americas
Countries and Regions: United States
Time zones: Central Time
Root Password: *****)
Override NFS version 4: [no]

```

To examine the configured zone, we can do the following.

```

global# zonecfg -z wasdmzone info
zonepath: /export/zones/wasdmzone
autoboot: false
pool:
inherit-pkg-dir:
  dir: /lib
inherit-pkg-dir:
  dir: /platform
inherit-pkg-dir:
  dir: /sbin
inherit-pkg-dir:
  dir: /usr
inherit-pkg-dir:
  dir: /opt/IBM/WebSphere
net:
  address: 192.168.1.200/24
  physical: cel

```

In the output above, the zone has been created as a Sparse Root Zone because it has one or more “inherit-pkg-dir” in /lib, /platform, /sbin, /usr, and /opt. For additional information, refer to *Solaris System Administration Guide: Solaris Zones* (see *Whole Root Zone and Sparse Root Zone*).

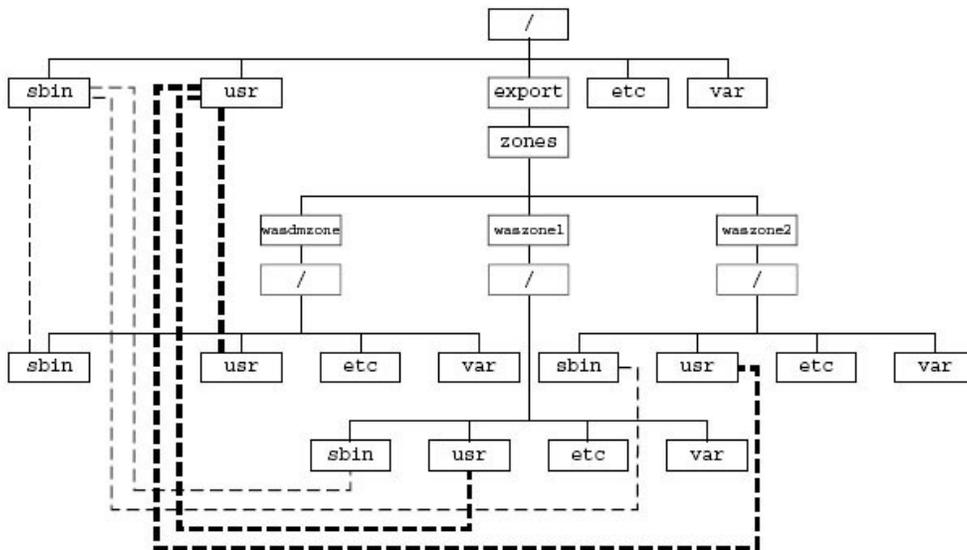
The zone's file reside in the “zonepath” (e.g. /export/zones/wasdmzone) that was defined during the zone configuration process. This root seen from the global zone is the same as logging into the non-global zone and doing “ls /” there.

```

global# ls /export/zones/wasdmzone/root

```

Figure 5 below shows the file system layout. The dashed lines are to depict the inherited file systems that are also known as the LOpBack File System (LOFS). The shared directory /opt/IBM/WebSphere is not depicted in the figure, but consider it similar to /usr.



**Figure 5: Solaris Zone and the File System Layout**

Once the zone is configured, and installed log in to it. Here is an example to login to the zone's system console.

```
global# zlogin -C wasdmzone
Last login: Mon Nov 28 18:15:27 on console
Sun Microsystems Inc. SunOS 5.10 Generic January 2005
```

As described in the WAS Installation section on profiles, we have modified the `wasprofile.properties` to point to our user defined directory `/opt/WASProfiles` rather than the default path `${was.install.root}`. We need to create the following three sub-directories for WAS profiles to be successfully created.

```
wasdmzone# mkdir -p /opt/WASProfiles/profile
wasdmzone# mkdir -p /opt/WASProfiles/logs
wasdmzone# mkdir -p /opt/WASProfiles/properties
```

Stay log in in the zone and start up the profile creation utility (wizard) called `pctSolaris.bin`. Choose between the Deployment Manager and Application Server profile. In this example, we will create the Deployment Manager profile.

```
wasdmzone# cd /opt/IBM/WebSphere/AppServer/bin/ProfileCreator
wasdmzone# ./pctSolaris.bin

or

wasdmzone# /opt/IBM/WebSphere/AppServer/bin/wasprofile.sh -create \
  -profileName "Dmgr01" \
  -profilePath "/opt/WASProfiles/profile/Dmgr01" \
  -templatePath "/opt/IBM/WebSphere/AppServer/profileTemplates/dmgr" \
  -hostName "wasdmzone" \
  -nodeName "wasdmzoneCellManager01" \
  -cellName "wasdmzoneCell01" \
  -dmgrHost "localhost" -dmgrPort "8879" \
  -federateLater -OS_NAME "SunOS" \
  -winserviceCheck "false" \
  -winserviceAccountType "localsystem" \
  -winserviceUserName "root" \
  -winservicePassword "*****" \
  manual -portsFile "/portdef.props"
```

Start the deployment manager.

```
wasdmzone# /opt/WASProfiles/profile/Dmgr01/bin/startManager.sh
```

Verify the Deployment Manager's functionality. Bring up a Web Browser and go to:

```
http://192.168.1.200:9060/admin
```

Repeat the above steps to create `waszone1` and WebSphere Application Server profile `AppSrv01`.

```
global# vi /waszone/waszone1.cfg
#
# Script to create WAS Node1 Zone
#
create
set zonepath=/export/zones/waszone1
set autoboot=true
add net
  set physical=cel
  set address=192.168.1.201/24
end
add inherit-pkg-dir
  set dir=/opt/IBM/WebSphere
end
verify
commit
```

```

#
# End of Script
#
global# zonecfg -z waszone1 -f /waszone/waszone1.cfg
global# zoneadm -z waszone1 install
Preparing to install zone <waszone1>.
Creating list of files to copy from the global zone.
Copying <2579> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <1032> packages on the zone.
global# zoneadm -z waszone1 boot
global# zlogin -C waszone1

```

Configure the zone with necessary information such as language, locale, timezone, hostname, name service, root password, etc. just like doing a “Sys Config” for a fresh Solaris environment.

```

waszone1# mkdir -p /opt/WASProfiles/profile
waszone1# mkdir -p /opt/WASProfiles/logs
waszone1# mkdir -p /opt/WASProfiles/properties
waszone1# cd /opt/IBM/WebSphere/AppServer/bin/ProfileCreator
waszone1# ./pctSolaris.bin

```

OR

```

waszone1# /opt/IBM/WebSphere/AppServer/bin/wasprofile.sh -create \
    -profileName "AppSrv01" \
    -profilePath "/opt/WASProfiles/profile/AppSrv01" \
    -templatePath "/opt/IBM/WebSphere/AppServer/profileTemplates/default" \
    -hostName "waszone1" \
    -nodeName "waszone1CellManager01" \
    -cellName "waszone1Cell101" \
    -dmgrHost "localhost" -dmgrPort "8879" \
    -federateLater -OS_NAME "SunOS" \
    -winserviceCheck "false" \
    -winserviceAccountType "localsystem" \
    -winserviceUserName "root" \
    -winservicePassword "*****" \
    manual -portsFile "/portdef.props"
waszone1# /opt/WASProfiles/profile/AppSrv01/bin/startServer.sh server1

```

Make sure the host id's are defined in the zone. If they are already defined in a name service (e.g. DNS, NIS) and the zone is configured to utilize the service, this step is not necessary.

Note: Zone cloning feature is not available at this time, so we will repeat the process here.

Repeat the above steps to create waszone2 and WebSphere Application Server profile AppSrv01 (It is acceptable to have the same profile name as waszone1).

```

global# vi /waszone/waszone2.cfg
#
# Script to create WAS Node2 Zone
#
create
set zonepath=/export/zones/waszone2
set autoboot=true
add net
    set physical=cel
    set address=192.168.1.202/24
end
add inherit-pkg-dir
    set dir=/opt/IBM/WebSphere
end
verify
commit
#
# End of Script
#
global# zonecfg -z waszone2 -f /waszone/waszone2.cfg

```

```

global# zoneadm -z waszone2 install
Preparing to install zone <waszone2>.
Creating list of files to copy from the global zone.
Copying <2579> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <1032> packages on the zone.
global# zoneadm -z waszone2 boot
global# zlogin -C waszone2

```

Configure the zone with necessary information such as language, locale, timezone, hostname, name service, root password, etc. just like doing a “Sys Config” for a fresh Solaris environment. **Note:** A zone has its own identity – a virtual system that has similar characteristics of a physical system. Two zones can have two different time zones, root passwords, languages, etc.

```

waszone2# mkdir -p /opt/WASProfiles/profile
waszone2# mkdir -p /opt/WASProfiles/logs
waszone2# mkdir -p /opt/WASProfiles/properties
waszone2# cd /opt/IBM/WebSphere/AppServer/bin/ProfileCreator
waszone2# ./pctSolaris.bin

```

Or

```

waszone2# /opt/IBM/WebSphere/AppServer/bin/wasprofile.sh -create \
    -profileName "AppSrv02" \
    -profilePath "/opt/WASProfiles/profile/AppSrv01" \
    -templatePath "/opt/IBM/WebSphere/AppServer/profileTemplates/default" \
    -hostName "waszone2" \
    -nodeName "waszone2CellManager01" \
    -cellName "waszone2Cell101" \
    -dmgrHost "localhost" -dmgrPort "8879" \
    -federateLater -OS_NAME "SunOS" \
    -winserviceCheck "false" \
    -winserviceAccountType "localsystem" \
    -winserviceUserName "root" \
    -winservicePassword "*****" \
    manual -portsFile "/portdef.props"
waszone2# /opt/WASProfiles/profile/AppSrv01/bin/startServer.sh server1

```

Make sure the host id's are defined in the zone. If they are already defined in a name service (e.g. DNS, NIS) and the zone is configured to utilize the service, this step is not necessary.

```

waszone2# cat /etc/hosts
#
# Internet host table
#
127.0.0.1    localhost
192.168.1.202 waszone2    loghost
192.168.1.200 wasdmzone
192.168.1.201 waszone1

```

Verify that the WAS nodes are up and running. Go to the two URL's below. This is through WAS internal HTTP transport access. The next section describes how Sun Web Server can be configured in a zone to handle HTTP traffic.

```

http://waszone1/PlantsByWebSphere
http://waszone2/PlantsByWebSphere

```

To have the the two WAS nodes federated to the Deployment Manager (join the cell), log in to each zone or do the following from the global zone.

```

global# ssh waszone1 \"/opt/WASProfiles/profile/AppSrv01/bin/addNode.sh \
    wasdmzone -includeapps -includebuses"
global# ssh waszone2 \"/opt/WASProfiles/profile/AppSrv01/bin/addNode.sh \
    wasdmzone -includeapps -includebuses"

```

## IV. Create another non-global zone and install Sun Java System Web Server

The zone configuration file for the Web Server is slightly different from the WebSphere zones since it does not need to inherit the WebSphere installation directories. Instead, we need to make this zone to have its own `/usr` directory because IBM WebSphere Web Server Plugins installation needs to create symbolic links to `/usr/lib` for `gsk4` and `gsk7` shared libraries. If you do not do this in creating the non-global zone, Web Server start up will fail after installation.

**Note:** If `/usr` is mounted as read-only by default, an alternate option is to physically copy the `gsk` shared libraries to `/usr/lib` in the global zone. We prefer the local `/usr` option as mentioned above. For example,

```
cp /opt/ibm/gsk*/lib/lib*.so /usr/lib
```

In the zone creation script for the Sun Web Server, we add “`remove inherit-pkg-dir dir=/usr`” line. For additional information, refer to [Solaris System Administration Guide: Solaris Zones](http://docs.sun.com/app/docs/doc/817-1592) (see *Whole Root Zone and Sparse Root Zone*). <http://docs.sun.com/app/docs/doc/817-1592>

```
global# vi /waszone/websvr1.cfg
#
# Script to create Zone to run Sun Web Server
#
create
set zonepath=/export/zones/websvr1
set autoboot=true
remove inherit-pkg-dir dir=/usr
add net
    set physical=ce0
    set address=10.1.197.59/24
end
verify
commit
#
# End of Script
#
global# zonecfg -z websvr1 -f /waszone/websvr1.cfg
global# zoneadm -z websvr1 install
Preparing to install zone <websvr1>.
Creating list of files to copy from the global zone.
Copying <129341> files to the zone.
Initializing zone product registry.
Determining zone package initialization order.
Preparing to initialize <1032> packages on the zone.
Initializing package <479> of <1032>: percent complete: 46%
global# zoneadm -z websvr1 boot
global# zlogin -C websvr1
```

Configure the `websvr1` zone with necessary information such as language, timezone, hostname, root password, etc. just like doing a “sys config” for a fresh Solaris environment.

Download and install the Sun Java System Web Server 6.1 SP2.

```
websvr1# cd SUN_WEBSVR_DIST
websvr1# ./setup
.... follow the defaults mostly ....
```

Start the Sun Web Server admin server.

```
websvr1# /opt/SUNWwebsvr/https-websvr1/start
```

Verify the existence of the Sun Web Server by going to its URL's.

```
http://10.1.197.59
```

Start the Sun Web Server admin server.

```
websvr1# /opt/SUNWwbsvr/https-admserv/start
```

Verify the Sun Web Server Console by going to the URL's below.

```
http://10.1.197.59:8888
```

Try accessing the WebSphere URL: "PlantsByWebSphere" and you will see an error message since the Web Server has not been configured to communicate with WAS.

```
http://10.1.197.59/PlantsByWebSphere
```

## V. Install WebSphere Plug-in on Sun JS Web Server

Install the WebSphere Web Server Plug-in.

```
websvr1# /opt/SUNWwbsvr/https-websvr1/stop
websvr1# cd ${WAS_SW_DIST}/plugin
websvr1# export DISPLAY=${WORKSTATION}:1.0
websvr1# ./install
```

Select "Sun ONE Web Server 6.0 or Sun Java System Web Server V6.1".

Since we are installing this Web Server in one zone while the WAS profiles are in different zones, we'll select "Web server machine (remote)".

Web Server plug-ins installation location: /opt/WebSphere/Plugins (Keep in mind that we do not have write privileges in /opt/IBM/WebSphere that is in the global zone.)

Sun Java System obj.conf file: /opt/SUNWwbsvr/https-websvr1/config/obj.conf

Sun Java System magnus.conf file: /opt/SUNWwbsvr/https-websvr1/config/magnus.conf

Port: 80

Web server definition: webserver1 (default)

Web server plugin-cfg.xml file: /opt/WebSphere/Plugins/config/webserver1/plugin-cfg.xml

Host name or IP address for the Application Server: 192.168.1.101

The plug-in takes up about 200 MB.

Restart the Web Server with the WebSphere plug-in installed.

```
websvr1# /opt/SUNWwbsvr/https-websvr1/start
```

Verify the WebSphere Plug-in for Sun Web Server by going to its URL's.

```
http://10.1.197.59/PlantsByWebSphere
```

## VI. Remove Unwanted Zones

The following commands allow you to inquire the status of existing zones on the system and remove unwanted zone (e.g. zone1). You must execute these commands from the global zone.

```
global# zoneadm list -cv
global# zoneadm -z zone1 halt
global# zoneadm -z zone1 uninstall
global# zonecfg -z zone1 delete
```

## Conclusions and Summary

Solaris 10 has innovative features that enable customers, like WebSphere users, to develop new ways to solve business problems and to drive down operational costs. Key features in Solaris 10 are as follows:

- Solaris Containers (Zones, Resource Control)
- Overall system performance over previous Solaris and Linux OS'es
- Optimized TCP/IP Stack
- Dynamic Tracing (DTrace)
- User and Process Rights Management
- Availability on Intel/AMD processors

As we demonstrated in this document, Solaris Zones provide many benefits. They are manageable, lightweight and easy to implement. A zone can be configured and installed quickly. An inactive zone (i.e. Not booted) has no overhead on the system except disk space. Unlike other virtualization technologies, an active zone has virtually no overhead except for a few processes required for zone specific operations. The Zone technology is available on Solaris SPARC and IA-32/64 environments.

Solaris Zone provides a secure “sandbox” that includes:

- A virtual platform containing a unique root, shared user, and administrator-configured file systems — plus network interfaces, IPC objects, a console, devices, and resource management facilities.
- Standard system identity settings including host name, time zone, RPC domain, and locale.
- An independent name space including users, roles, and process IDs.
- Secure isolation from other zones enforced at the kernel level. A process in a Solaris Zone, even if compromised, cannot escalate privileges to compromise the system or another zone.
- Fault isolation that can restrict the propagation of software faults to a single zone. A non-global zone can reboot in only a few seconds.

Zone provides observability where the global zone administrator can see all activities in non-global zones (e.g. `ps -Zef` or `dtrace`).

Zones enable the standard Solaris interfaces and application environment, and do not impose a new ABI or API. Applications, such as WebSphere Application Server, that can execute without root privileges work correctly in a non-global zone. For more details, see: [http://developers.sun.com/solaris/articles/application\\_in\\_zone.html](http://developers.sun.com/solaris/articles/application_in_zone.html)

Zones provide isolated software environment for web facing applications, such as WebSphere Application Environment, and extend the container notion beyond the scope of Java Enterprise Edition (formerly known as J2EE). Web and application containers provide secure execution

environments for the respective components. Solaris Zones provide secure execution environment and virtualization for these containers – Web Containers and EJB Containers. Web Services deployment can be compartmentalized securely. By using the Solaris Container capabilities (i.e Solaris Zones combined with Solaris Resource Manager and Resource Pools), higher server utilization can be realized in these service deployments (see “*Solaris Containers--What They Are and How to Use Them*” by Menno Lageman, Sun Microsystems).

In summary, to get the maximum benefit from your IT investment, you must consider these innovative capabilities as part of your overall strategy to reduce server sprawl and increase resource utilization in a systemic and secure manner.

## **Additional Resources**

### Sun Manuals

#### ***System Administration Guide: Solaris Containers-Resource Management and Solaris Zones***

<http://docs.sun.com/app/docs/doc/817-1592>

#### ***Solaris Tunable Parameters***

<http://docs.sun.com/app/docs/doc/817-0404/>

### Sun Blueprints Articles

#### ***Solaris Containers--What They Are and How to Use Them***

<http://www.sun.com/blueprints/0505/819-2679.pdf>

#### ***Slicing and Dicing Servers: A Guide to Virtualization and Containment Technologies***

<http://www.sun.com/blueprints/1005/819-3734.pdf>

#### ***Restricting Service Administration in the Solaris™ 10 Operating System***

<http://www.sun.com/blueprints/0605/819-2887.pdf>

### Sun Whitepapers

#### ***Solaris™ Containers: Server Virtualization and Manageability***

[http://www.sun.com/software/whitepapers/solaris10/grid\\_containers.pdf](http://www.sun.com/software/whitepapers/solaris10/grid_containers.pdf)

### IBM Manuals

#### ***Installing Your Application Server Environment***

[ftp://ftp.software.ibm.com/software/webserver/appserv/library/v60/wasv600nd\\_gs.pdf](ftp://ftp.software.ibm.com/software/webserver/appserv/library/v60/wasv600nd_gs.pdf)

#### ***Setting Up the Application Serving Environment***

[ftp://ftp.software.ibm.com/software/webserver/appserv/library/v60/wasv600nd\\_env.pdf](ftp://ftp.software.ibm.com/software/webserver/appserv/library/v60/wasv600nd_env.pdf)

#### ***Installing V6.0.2 WebSphere Application Server on Solaris Systems***

<http://www-1.ibm.com/support/docview.wss?rs=180&context=SSEQTP&uid=swg21210054>

#### ***Sharing WebSphere Binaries (on zLinux)***

[http://www.ibm.com/servers/eserver/zseries/os/linux/pdf/sharing\\_webSphere\\_binaries.pdf](http://www.ibm.com/servers/eserver/zseries/os/linux/pdf/sharing_webSphere_binaries.pdf)

### IBM Redbooks and Redpapers

#### ***WebSphere Application Server V6 Planning and Design WebSphere Handbook Series***

<http://www.redbooks.ibm.com/abstracts/sg246446.html>

## **Other References**

#### ***Ade Rixon's Blog Entry "Running WebSphere (V5.x) In A Solaris Zone" on July 15, 2004***

<http://www.big-bubbles.fluff.org/blogs/bubbles/archives/000344.html>