Setting Up the Application Development Environment in Oracle Solaris 11.4

E61049-02
February 2024

ORACLE

Setting Up the Application Development Environment in Oracle Solaris 11.4,

E61049-02

Contents

Using This Documentation

- **Overview -** Describes setting up a development environment on the Oracle Solaris 11.4 operating system and also provides links to sources of detailed information for developers.
- **Audience -** Developers using a variety of programming languages including C, C++, Java, PHP, and Ruby.
- **Required Knowledge -** Readers of this guide should be familiar with basic experience in developing applications.

Product Documentation Library

Documentation and resources for this product and related products are available at http://www.oracle.com/pls/topic/lookup?ctx=E37838-01.

Feedback

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

1

Introduction to Setting Up an Application Development Environment in Oracle Solaris 11

Setting up an application development environment involves activities such as installing developer tools, installing and configuring web servers, installing databases and so on. Oracle Solaris 11 provides various packages and tools that enable you to develop applications. This guide covers the following topics:

- Installing Software Useful for Application Development
- Installing Oracle Developer Studio
- Installing Web and Application Servers
- Installing a Version Control System
- Installing a Database
- Debugging Applications
- Creating IPS Packages
- Configuring Boot Environments

Installing Software Useful for Application Development

The Oracle Solaris 11 OS provides various software packages that enable you to develop, debug, and maintain applications. The following packages are likely to be helpful when developing applications on the Oracle Solaris operating system.

- `developer/build/ant` installs Apache Ant
- `developer/build/automake` installs a makefile generator
- `developer/build/gnu-make` installs a utility for directing compilation
- `developer/build/meson` installs the Meson build configuration tool
- `developer/build/pkg-config` installs a software configuration query tool
- `developer/dtrace/toolkit` installs a collection of useful documented DTrace scripts
- `developer/debug/gdb` installs the GNU project debugger
- `developer/debug/mdb` installs the Solaris Modular Debugger, see Debugging Applications
- `developer/debug/valgrind` installs the Valgrind memory debugging tool (x86 platforms only)
- `developer/documentation-tool/openjade` installs the DSSSL engine for SGML documents

- `developer/gcc` installs the GNU Compiler Collection, including compilers for C, C++, Objective C, Fortran, and Go
- `developer/java/jdk` installs the Java Platform Standard Edition Development Kit
- `developer/lexer/flex` installs the Flex lexical analyzer
- `developer/llvm/clang` installs the LLVM compilers for C and C++
- `developer/rust/rustc` installs the Rust compiler
- `developer/ui-designer/glade` installs the GNOME UI designer
- `developer/vala` installs the Vala programming language
- `developer/versioning/git` installs the Git version control system
- `developer/versioning/mercurial` installs the Mercurial version control system
- `developer/versioning/subversion` installs the Subversion version control system
- `group/feature/amp` installs the AMP (Apache, MySQL, PHP) Deployment Kit for Oracle Solaris
- `group/feature/developer-gnu` installs the GNU Development Tools
- `library/libmemcached` installs an open source C/C++ client library and tools for the memcached server
- `runtime/perl-536` installs Perl
- `runtime/ruby` installs Ruby & RubyGems
- `runtime/tcl-8` installs Tcl, a portable scripting environment
- `web/java-servlet/tomcat` installs the Tomcat Servlet/JSP Container
- `web/php` installs the PHP server
- `web/proxy/squid` installs the Squid web proxy cache
- `web/server/lighttpd-14` installs the Lighttpd web server

> **Tip:**
>
> To view a comprehensive list of packages useful for development, type `pkg list -as 'developer/*'` in a terminal window.

Use the `pkg` command to install or update a software. For example, to install `Ruby`, you would run the following command:

```
$ pkg install runtime/ruby
```

For information about the `pkg` command, see pkg(1) and *Updating Systems and Adding Software in Oracle Solaris 11.4*.

Installing Oracle Developer Studio

Oracle Developer Studio consists of two suites of tools: a compiler suite and an analysis suite. The tools of each suite are designed to work together to provide an optimized development environment for the development of single, multithreaded, and distributed applications.

Oracle Developer Studio provides everything you need to develop C, C++, and Fortran applications to run in Oracle Solaris 10 or Oracle Solaris 11 on SPARC or x86 and x64 platforms, or in Oracle Linux on x86 and x64 platforms. The compilers and analysis tools are engineered to make your applications run optimally on Oracle Solaris systems.

The components of Oracle Developer Studio include:

- **IDE** – An integrated development environment for application development in a graphical environment. The Oracle Developer Studio IDE integrates several other Oracle Developer Studio tools and uses Oracle Solaris technologies such as DTrace.
- **C compiler** – Includes a C compiler, incremental link editor, and lint program.
- **C++ compiler** – Includes a full-featured C++ compiler and interval arithmetic library.
- **Fortran compiler** – Includes a full-featured environment and libraries for both f95 and f77.
- **dbx debugger** – An interactive, source-level, command-line debugging tool.
- **dmake make tool** – A command-line tool for building targets in distributed, parallel, or serial mode.
- **Math libraries** – A floating-point environment that is supported by software and hardware on SPARC and x86 system that run the Oracle Solaris OS.
- **OpenMP** – A portable, pragma-based parallel programming model for shared memory multiprocessor architectures. It is natively accepted and compiled by all three Oracle Developer Studio compilers.
- **Performance Analyzer** – A GUI and command-line tool for collecting and analyzing performance data.
- **Thread Analyzer** – A GUI and command-line tool for analyzing the execution of multithreaded programs and checking for a variety of multithreaded programming errors.
- **Oracle Performance Library** – A library of Oracle-specific extensions and features for using optimized, high-speed mathematical subroutines for solving linear algebra and other numerically intensive problems.

Oracle Developer Studio is freely available for production use on Oracle Solaris and Linux operating systems. It is available both as an IPS package and a tar file. You can download Oracle Developer Studio from the Oracle Developer Studio web site.

> **Note:**
>
> To download the software you must have an Oracle web account. If you do not have an Oracle web account, you can sign up for free.

To install Oracle Developer Studio IPS packages, you must have a key and a certificate. Visit the Oracle Package Repositories site and follow the instructions to install the key and certificate.

Installing Web and Application Servers

As a developer, you might need an application server to test your programs, deploy your applications, and run test scenarios. This section describes the servers that you can install.

- Apache HTTP Server
- Oracle iPlanet Web Server
- Oracle HTTP Server

By default, the Apache web server package (`web/server/apache-24`) is included in `solaris-minimal-server` installation package.

Apache HTTP Server

Apache is a widely used open-source web server. Apache HTTP Server version 2.4 is available as an IPS package for the Oracle Solaris 11 operating system. For more information on the Apache HTTP Server, see the Apache documentation.

Installing the Apache HTTP Server

1. Type the following command:

   ```
   $ pkg install web/server/apache-24
   ```

   > **Tip:**
   >
   > Alternatively, you can install the `group/feature/amp` package. This package contains Apache HTTP Server, MySQL database, and PHP.

2. Enable the server so that it listens to the incoming HTTP requests.

   ```
   $ svcadm -v enable /network/http:apache24
   ```

3. To verify that the web server works, open the http://localhost:80 link in a web browser.

   A valid web page should be displayed.

Oracle iPlanet Web Server

Oracle iPlanet Web Server is a high-performance web server that improves web security, enhances the end-user experience, and reduces the cost and complexity of deploying and managing web applications. It is available on the Oracle Solaris,

Windows, HP-UX, AIX, and GNU/Linux platforms. It supports the JSP and Java Servlet technologies, PHP, NSAPI, CGI, and ColdFusion.

Oracle iPlanet Web Server is available for download at the Oracle iPlanet Web Server website.

For installation instructions, see How to Install Oracle iPlanet Web Server.

Oracle HTTP Server

Oracle HTTP Server is the web server component for Oracle Fusion Middleware. It provides a listener for Oracle WebLogic Server and the framework for hosting static pages, dynamic pages, and applications over the web.

Oracle HTTP Server serves static content directly or through standard interfaces such as the **`WebDAV`** standard. It provides features such as single sign-on, clustered deployment, and high availability. You can use it as a proxy server, both forward and reverse. A reverse proxy enables content served by different servers to appear as if coming from a single server.

For more information, see the Oracle HTTP Server website.

Installing a Version Control System

Oracle Solaris does not have a proprietary version control system. However, you can install and configure an open-source version control system. Some of the open-source version control systems that are available as IPS packages are the following:

- Git - `developer/versioning/git`
- Mercurial - `developer/versioning/mercurial`
- SCCS - `developer/versioning/sccs`
- Subversion - `developer/versioning/subversion`

Installing a Database

The Oracle Solaris operating system supports Oracle, MySQL, Berkeley DB, PostgreSQL, and SQLite databases. Open-source databases might not be available as an IPS package and you have to download these software from third party websites.

The following sections provide brief installation information about Oracle and MySQL databases.

- Oracle Database
- MySQL Database

Oracle Database

Oracle Database is available in different editions, which can scale from small to large single servers and clusters of servers. To know the high-level information about different Oracle Database options, see Latest Oracle Database. Documentation links are at the bottom of the page.

Oracle Instant Client enables applications to connect local or remote Oracle Databases for development and production. You can install Oracle Instant Client on Oracle Solaris 11.3 or later versions using the command:

```
$ pkg install database/oracle/instantclient
```

For more information, see Oracle Instant Client.

> **Note:**
>
> As a pre-requisite to install the Oracle Database on Oracle Solaris, you must install the `group/prerequisite/oracle/oracle-rdbms-server-12-1-preinstall` group package. This group package has all the required packages.

MySQL Database

The MySQL database is available as an IPS package. For information about using MySQL, see the *MySQL 8.0 Reference Manual*.

How to install the MySQL IPS Package

1. Type the following command:

   ```
   $ pkg install database/mysql-80
   ```

   After the installation is complete, you must start the database.
2. Start the database:

   ```
   $ svcadm enable mysql
   ```

3. Type `mysql` in a terminal window to access the `mysql>` prompt.

   For example:

   ```
   mysql> show databases;
   Database
   information_schema
   mysql
   test
   3 rows in set (0.01 sec)

   mysql> quit;
   Bye
   ```

   For more information about tuning ZFS for database, see "Tuning ZFS for Database Products" in the *Oracle Solaris 11.4 Tunable Parameters Reference Manual*.

Debugging Applications

When you are developing an application, debugging the application becomes an important part of application development. Debugging tools are available that enable you to debug complex software systems effortlessly. The commonly used debugging tools are as follows:

- `mdb` – This extensible, general purpose debugging tool for the Oracle Solaris operating system enables you to debug at assembly language level. By using mdb, you can debug the Oracle Solaris kernel and associated device drivers and modules. Along with mdb you can use libumem to find memory leaks and bugs in applications and libadimalloc to find buffer overflows and bad pointers on machines with Silicon Secured Memory (SSM) such as the Oracle SPARC M7, T7, S7, M8, and T8 servers. For information, see:
  - *What's New in Oracle Solaris 11.4*
  - Library-Level Dynamic Memory in the *Oracle Solaris 11.4 Programming Interfaces Guide*
  - *Oracle Solaris Modular Debugger Guide*
  - https://blogs.oracle.com/jwadams/entry/debugging_with_libumem_and_mdb
  - https://web.archive.org/web/20080212075502/http://developers.sun.com/solaris/articles/libumem_library.html
- gdb – You can use the GNU debugger, commonly called as gdb to debug applications, gdb allows you to analyze and debug programs at runtime. It also allows you add specific breakpoints to examine the program. For more information, see GNU Debugger.
- dtrace – This comprehensive dynamic tracing framework for the Oracle Solaris OS enables you to concisely answer questions about the behavior of the operating system and user programs. DTrace can help you to identify performance issues and bugs in applications. For more information, see the Oracle Solaris 11.4 DTrace (Dynamic Tracing) Guide.
- Oracle Developer Studio provides the following tools for debugging applications.
  - dbx – This interactive, source-level debugging tool is used to run a program in a controlled manner. It gives you complete control of dynamic execution of a program such as collecting performance and memory usage data, monitoring memory access, and detecting memory leaks. You can also use the GUI version of dbx by running the dbxtool command. For more information, see Oracle Developer Studio 12.6: Debugging a Program with dbx and Oracle Developer Studio 12.6: dbxtool Tutorial.
  - discover – This command line utility helps to detect memory access errors in your code. See Memory Error Discovery Tool (discover).
  - Code Analyzer – This integrated set of tools helps you to ensure application reliability and security by detecting application vulnerabilities, including memory leaks and memory access violations. For more information, see *Oracle Developer Studio 12.6: Code Analyzer User's Guide* and Oracle Developer Studio 12.6: Code Analyzer Tutorial.
  - Collector and Performance Analyzer – These tools perform statistical profiling of a wide range of performance data and tracing of various system calls, and relate the data to program structure at the function, source-level, and instruction levels. For more information, see Oracle Developer Studio 12.6: Performance Analyzer and Oracle Developer Studio 12.6: Performance Anaylzer Tutorials.

- Thread Analyzer – It can detect hard to find complex parallel programming errors. Thread Analyzer supports POSIX, OpenMP, and Oracle Solaris threads. For more information, see the *Oracle Developer Studio 12.6: Thread Analyzer User's Guide*.

Creating IPS Packages

In the Oracle Solaris 11 operating system, you should deliver the software you develop as Image Package System (IPS) packages. IPS packages are installed and updated from IPS repositories.

IPS is a framework for complete software life cycle management including packaging, installing, upgrading, and removing software. You must deliver software as IPS packages to obtain the following benefits:

- Automatically calculate sufficient correct dependencies on other software
- Automatically install or update dependency software as needed
- Automatically refresh or restart necessary system services to perform tasks such as software configuration and automatically starting other dependent services
- Form a single package, automatically install only the components appropriate for the existing system architecture, or other variants
- Make updates in a new boot environment without modifying the existing boot environment
- Enable users to easily verify whether the software installation is correct

For more information about creating IPS packages, see *Packaging and Delivering Software With the Image Packaging System in Oracle Solaris 11.4*.

Configuring Boot Environments

A boot environment is a bootable instance of the Oracle Solaris operating system image along with any other software packages installed in that image. You can maintain multiple boot environments on your systems, and each boot environment can have different software versions installed. You can backup the current boot environment, you can update software without any risk of loss of data or the system environment. You can also update a boot environment that is currently not active. Use the beadm(1M) utility to create and manage boot environments.

You can use zones which can help you to maintain multiple boot environments with different versions of software installed. Zones can be a huge advantage when you do not have to reboot the system to access another environment. You can also use kernel zones to maintain different versions of OS. Oracle Solaris allows you to have several non-global zones, several kernel zones, and non-global zones inside a kernel zone. You can easily `ssh` to one of the zones without rebooting the physical machine.

For more information about boot environments, see *Creating and Administering Oracle Solaris 11.4 Boot Environments*.

2

Creating a Virtual Development Environment

A virtual development environment (VDE) is an application development environment (ADE) set up on a virtual machine. By using virtual machines, you can work with multiple Operating Systems (OS) without the need for a separate hardware for each OS. Oracle Solaris offers multiple options for setting up and configuring virtual environments. This document states how you can choose, set up, and configure the virtual environment. Because a virtual machine acts like a physical machine for the purposes of hosting an ADE, the process for setting up the ADE is the same for virtual or physical machines. This chapter provides guidelines for choosing, setting up, and configuring a virtual environment to host your ADE.

Virtualizing Your Development Environment

Virtualizing your development environment through Oracle Solaris has multiple benefits:

- Saving downtime – A VDE helps you reduce downtime by enabling you to develop your application in a virtual machine (VM) that is backed up and easily restored. A VM that is backed up on one system can be restored on a different system.
- Protecting hardware – Because VMs are virtual, developing applications on VMs cannot damage your hardware, which enables you to perform risky procedures safely. You can back up and restore easily without any damage to the underlying physical machine.
- Simulating production environment – Setting up a virtual development environment mimicking your production environment helps ensure that testing and developing your applications will produce useful results.
- Security – Strong security features mean you can use your VDE without compromising on security.
- Multiple OS versions – Virtualizing the development environment allows you to work with different versions of operating systems on a single physical machine.

Choosing a Virtualization Option

Before you can install your ADE, you need to plan and set up the virtualized environment that will host the ADE. One important consideration is the balance between isolation and flexibility.

The following table describes the virtualization models available for Oracle Solaris.

Table 2-1 Virtualization Models Available for Oracle Solaris

Virtualization Model	Virtualization Technology	Description
Operating System Virtualization	• Oracle Solaris Zones • Oracle Solaris Kernel Zones	Each environment contains what appears to be a private copy of the OS in a container. The OS virtualization model provides near-native performance and flexibility, and has a much smaller disk, RAM, and CPU footprint than either virtual machines or physical machine.
Virtual Machines	• Oracle VM Server for SPARC systems • Oracle VM Server for x86 systems • Oracle VM Virtual Box • Oracle Linux Virtualization Manager	By using virtual machines, you can run multiple OS instances with a single set of hardware resources. Each virtual machine that you create runs its own OS. A software or firmware hypervisor creates the illusion that each guest OS instance is running on its own system. Although virtual machines provide less resource flexibility than a physical machine that uses OS virtualization, they provide more isolation.
Physical Domains	Oracle SPARC Enterprise M-Series servers	Hardware partitions provide physical separation between the running OS and its separate set of resources and power. Because this model does not use a hypervisor, it provides bare-metal performance. This virtualization model provides the most isolation but it is much less flexible with resource configuration than either the virtual machine or OS virtualization model.

The following diagrams illustrate some of the virtualization models described in the table.

Figure 2-1 OS Virtualization by Using Oracle Solaris Kernel Zones

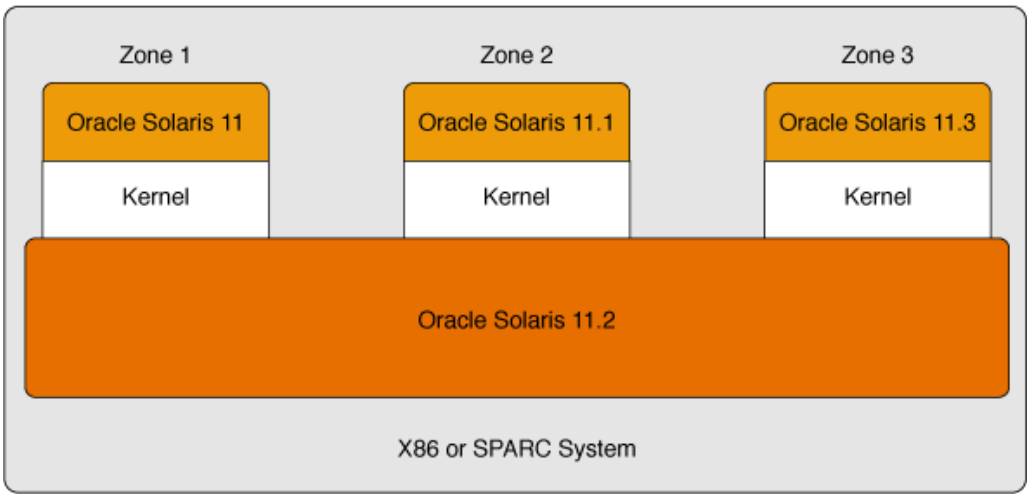


Figure 2-2 Using Virtual Machines on an Oracle SPARC System

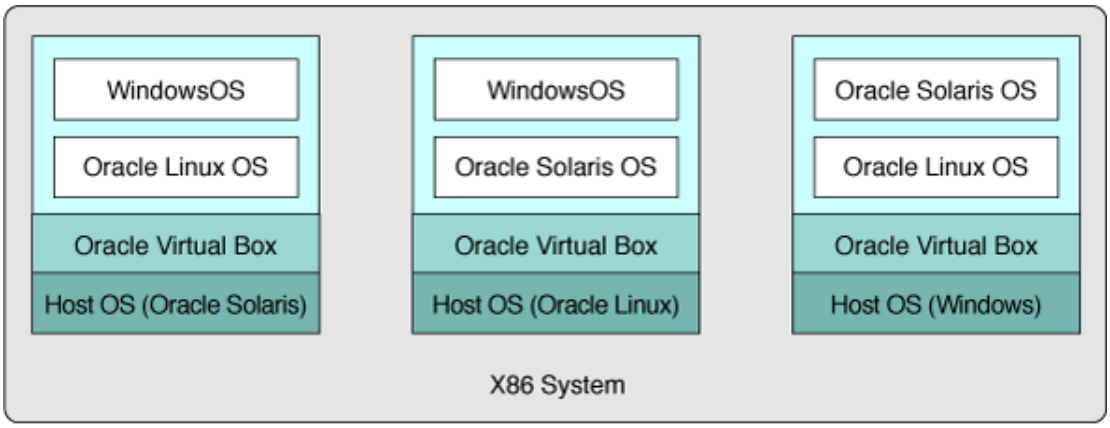
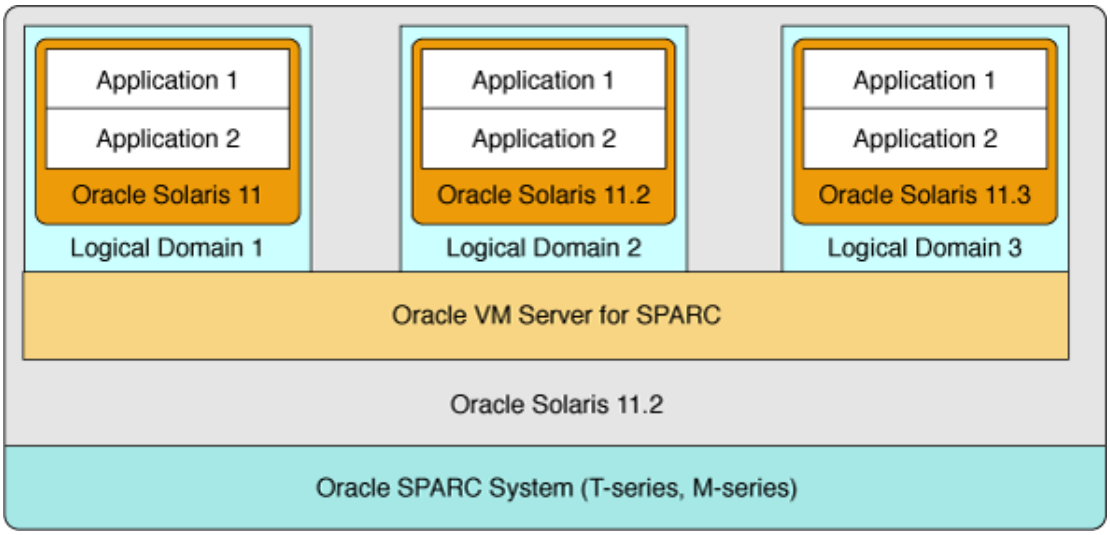


Figure 2-3 Using Virtual Machines on an Oracle x86 System



Choosing a Virtualization Technology

Oracle Solaris provides virtualization technologies that you can use in various virtualization models. This section focuses on compute virtualization relevant to creating a virtual development environment. For more information about how virtual environments are managed in a network, see *Introduction to Oracle Solaris 11.4 Virtual Environments*.

Table 2-2 Using Oracle Solaris Virtualization Technology in Your Environment

Virtualization Technology	Description	Sample Usage Scenarios	Using the Technology
Oracle Solaris Zones	A virtualized operating system environment created within a single instance of the Oracle Solaris operating system.	• Provide better resource management and control by installing similar applications in a common zone. • Balance workloads by logically separating the system. • Test migration scenarios for applications.	• *Introduction to Oracle Solaris Zones* • *Oracle Solaris Zones Configuration Resources* • *Creating and Using Oracle Solaris Zones*
Oracle Solaris Kernel Zones	A zone with its own kernel and OS installation separate from the global zone.	• Install multiple Oracle Solaris OS versions on a single physical machine. • Test migration scenarios for applications. • Require enhanced security and more isolation.	• *Introduction to Oracle Solaris Zones* • *Creating and Using Oracle Solaris Kernel Zones*
Oracle VM Server for SPARC system	Hypervisor-based virtualization for SPARC servers.	Deploy and configure different versions of the Oracle Solaris OS on SPARC T-Series and SPARC MSeries servers.	Oracle VM Server for SPARC
Oracle VM Server for x86 system (Xen)	Hypervisor-based virtualization for x86 based servers.	Deploy and configure x86 system with heterogeneous operating systems, including the Oracle Solaris 10 OS and the Oracle Solaris 11 as OS guests.	Oracle VM Server for x86 and Oracle VM Manager
Oracle VM VirtualBox	Hosted workstation and server virtualization for x86 based systems.	Develop and test software on different OS platforms. For example, you can install Oracle Solaris and Oracle Linux on a Windows system.	Oracle VM VirtualBox

Table 2-2 (Cont.) Using Oracle Solaris Virtualization Technology in Your Environment

Virtualization Technology	Description	Sample Usage Scenarios	Using the Technology
Physical domains (Also called as hardware partitions)	Electrical isolation into domains on Oracle SPARC M-Series servers.	Create multiple physical server entities that work independently of each other on a single hardware.	Oracle SPARC M-Series Servers
Oracle Linux Virtualization Manager	Server virtualization management platform for x86 based servers.	Deploy, configure, monitor, and manage an environment with enterprise-grade performance and support from Oracle.	Oracle Linux Virtualization Manager

Virtualization Tools

The following tools simplify and speed up installing and managing your virtual environment:

- **Oracle Solaris VM Templates** – Facilitate fast installation of Oracle Solaris virtual environments. The following types of Oracle Solaris VM templates are available:
  - Oracle VM Templates for SPARC system
  - Oracle VM Templates for x86 system
  - Oracle VM Templates for Oracle VM VirtualBox

For more information, see the Oracle Solaris VM Templates product page.