ORACLE
PEOPLESOFT

An Oracle Red Paper

December 2011

PeopleSoft Oracle Virtual Machine Templates Development and Customization Guide

ORACLE

Introduction

This Red Paper is a practical guide for technical users, installers, system administrators, and programmers who intend to leverage Oracle Virtual Machine (OVM) as a deployment infrastructure for PeopleSoft Applications.

The purpose of this document is to provide insight into the way in which PeopleSoft VM Templates are constructed. This will help you customize and extend the delivered templates. In addition this document can provide guidance for the creation of VM templates to run on other hypervisors.

Keep in mind that content within this document may be inconsistent with the version of the VM template with which you are working. Furthermore, the content herein may be based upon a different version of PeopleSoft than that with which you are working. For the latest features and most accurate information downloading the most recent version of templates from Oracle Software Delivery Cloud is strongly recommended. Consult the revision history later in this document for additional information.

Disclaimer

This material has not been submitted to any formal Oracle test and is published *as is.* It has not been the subject of rigorous review. Oracle assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by Oracle for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Information in this red paper was developed in conjunction with use of the products specified and is limited in application to those specific hardware and software product versions.

Structure of This Red Paper

This red paper provides guidance for development and customization of PeopleSoft Oracle VM (OVM) Templates. A description of the PeopleSoft OVM templates and how they were constructed is described at the beginning of this document. An abstract description of how templates can be created independent of the Oracle VM toolset is then provided. Finally, this document introduces the concept of template customization and how this can be achieved with the OVM templates created by PeopleSoft.

Oracle updates this document as needed so that it reflects the most current feedback from the field. Therefore, the structure, headings, content, and length of this document may vary with each posted version. To see if the document has been updated since you last downloaded it, compare the date of your version to the date of the version that is posted on Oracle Technical Network (OTN).

Related Materials

This paper draws upon existing documentation made available elsewhere. The table below provides links to relevant material referred to later in this document.

TABLE 1. RELATED MATERIALS

DOCUMENT	SOURCE
PeopleSoft VM Template Overview (white paper)	http://download.oracle.com/docs/cd/E16260_01/psft/html/docset.html
PeopleSoft Enterprise Hosted PeopleBooks	http://download.oracle.com/docs/cd/E17566_01/epm91pbr0/eng/psbooks/psft_homepage.htm
Oracle VM Template Developer's Guide	http://www.oracle.com/us/technologies/027000.pdf

TABLE 1. RELATED MATERIALS

DOCUMENT	SOURCE
Oracle VM Template Builder	http://www.oracle.com/technetwork/server-storage/vm/ovmtb-093027.html
My Oracle Support (MOS)	http://support.oracle.com/
Oracle Technical Network (OTN)	http://www.oracle.com/technetwork
Oracle VM VirtualBox Manual	http://www.virtualbox.org/manual
Template Deployment Guide	Accessible from the OTN page from which the template is downloaded. README documents within the template distribution will provide further information.

Overview

PeopleSoft Oracle VM Templates comprise a pre-packaged, ready-to-run software application. All third party products required by a basic PeopleSoft deployment are contained within the template distribution. The template distribution includes a demo grade Application[1] database. This template distribution is useful for getting a quick start environment running. However, most environments require customization and maintenance. This document provides insight into how the delivered template can be developed further to implement additional demands. This is achieved by providing a detailed explanation of how the template has been constructed. Understanding the mechanism by which the template distribution was created helps understand how to create your own template for Oracle VM or other virtualization platforms.

Who Should Read This Paper?

This document is a practical guide to understanding the internals of PeopleSoft Oracle VM templates. As such, it is written with the assumption that the reader is familiar with:

- Administration and provisioning of PeopleSoft environments
- Administration of the Linux Operating System
- Virtualization and Oracle VM in particular

This guide is intended for System Administrators seeking to better understand the way in which PeopleSoft Oracle VM Templates are constructed. Operational or System Architects seeking to customize and extend the PeopleSoft templates delivered by Oracle should read also this document. This will provide the necessary insight into understanding where the customization touch points are located within the template and the necessary steps to apply organization-specific changes to the delivered templates.

In addition, readers seeking to build their own templates from scratch should also consult this document. This will help understand the steps required to create a virtualized PeopleSoft appliance.

This document contains a significant amount of material. It has been written in such a way to acts as a reference. Care has been taken to make the sections as independent as possible. Consequently it is suggested that the reader begins with the table of contents and select the topics of most interest.

[1] Application refers to PeopleSoft Applications such as HCM, FSCM etc.

Before You Begin

The information provided in this guide can be of academic value without having direct access to an Oracle VM environment. However, it is recommended that you have access to an Oracle VM environment that has imported PeopleSoft Oracle VM templates. If you haven't already done so it is suggested that you download and install Oracle VM from Oracle Software Delivery Cloud. This is a separate undertaking that is outside the scope of this document.

You will be required to have administrator privileges on hardware that you use in order to implement most of the steps described in this document. For this reason it might be desirable to create an isolated environment that is intended for creating or customizing VM templates. Frequently referred to as a *sandbox,* an isolated environment of this nature can be restricted for development or demonstration purposes.

Common Terms

The following table provides definitions for some of the common terms that are used in this guide. If you are unfamiliar with some of the more technical material contained herein additional trips to external information sources such as wikipedia.org may be required.

TABLE 2. COMMON TERMS

TERM	DEFINITION
Bare-metal	Refers to a conventional non-virtual OS.
DHCP	Dynamic Host Configuration Protocol is a mechanism by which a host typically gets its IP address from another server on the network. This means that a host's IP address should be considered dynamic and can potentially change at any time.
IPC	Inter Process Communication resources are used by Tuxedo based programs like PeopleTools to implement queuing and exchange data between processes via shared memory. IPC values are set as kernel parameters on UNIX systems.
JeOS	Pronounced as 'juice' Just enough OS is a minimal operating system that contains just enough resources and software packages and configuration to achieve the task for which it has been intended. Normally no unnecessary software packages or disk space is included.
OL	Oracle Linux is a Red Hat Enterprise Linux distribution released by the Oracle Unbreakable Linux program.
OVM (Oracle Virtual Machine)	Oracle VM is a server based Virtualization solution based upon the open source Xen hypervisor. Oracle VM templates are delivered as pre-packaged software components to run on Oracle VM.
PIA	PeopleSoft Pure Internet Architecture is one of the major architectural components in a PeopleSoft environment. It runs a number of J2EE servlets across a designated set of web applications. These web applications communicate with the database through the PeopleTools Application Server.

TABLE 2. COMMON TERMS

TERM	DEFINITION
POSIX	Portable Operating System Interface for UNIX, is a standard program interface to which most variants of UNIX and Linux adhere.
RPM	Red Hat Package Manager is a system for managing software packages installed on Red Hat Linux systems. The noun 'RPM' also refers to the distribution mechanism for the software packages. An RPM can be thought analogous to a zip archive with some addition meta data and content extraction directives. Consult rpm.org for ample information about RPM.
Sandbox	Isolated development environment in which the users have limited access to the network. Conventional security restrictions imposed upon users can be relaxed.
VM Template	A Virtual Machine (VM) Template can be thought of as a runtime environment packaged as an archive similar to a zip or tar file. Virtual Machine instances are created from VM Templates. The process of creating a running VM from a template is achieved by instantiating with real values such as a host name, IP address and other configuration settings that are required to participate in a running environment. VMs and VM templates are often referred to by different terminology depending upon the virtualization platform on which they are intended to run. Terms such as virtual machine images and virtual appliances are frequently encountered.
Xen	Open source virtualization platform upon which Oracle VM is based. The functionality of Oracle VM Server closely resembles the way in which virtualized servers are managed. Concepts such as server pools, Oracle VM Manager and facilities such as live migration are delivered by the management layer that Oracle VM adds on top of Xen.

PeopleSoft VM Template Overview

This section provides an overview of the PeopleSoft Oracle VM templates and explains the purpose of delivering software in this manner. The functionality included in the PeopleSoft Oracle VM templates is also discussed.

Purpose and Intent

The goal of the PeopleSoft OVM template is twofold. These are described as follows.

1. Demo and Reference Environment Delivery

VM templates typically contain a demo PeopleSoft Application environment combined with the PeopleTools runtime on which it depends. By providing minimal networking details[2] a fully functional PeopleSoft demo environment can be started in less than an hour. This replaces the effort required to install the operating system, software applications and configure the environment. The demo environments deployed can be provided to functional teams to assess the feature and changes offered in the newer release of the PeopleSoft application. This is the primary value proposition of the demo templates discussed here.

The demo environment can also be used by system administrators to identify new third parties that have been added or runtime requirements that have been introduced. For example, a PeopleTools 8.49 customer may use a VM template containing PeopleTools 8.51 to plan their deployment strategy of the new release. This permits experimentation with newer features that come pre-packaged in the VM. For example, the Decoupled PS Home feature was introduced with PeopleTools 8.50. This feature relocates writeable content such as the domain configuration, logs, cache etc, outside PS_HOME. The template comes with this deployment by default. Using the VM template allows administrators to understand and analyze this new feature in a pre-configured environment. This pre-configured environment is set-up in a way that embodies best practice deployment from Oracle.

2. Starting Point for Post-Demo Environments

Although the VM functions as a demo environment out of the box there is nothing to prohibit the delivered VM from being extended and customized. The configuration and database content that is delivered is demo grade but the binary content that is shipped is production grade. It is ensured that the versions of the software bundled with the templates are those that are certified for the PeopleSoft Application release in question. This means that the template comprises an accurate starting point from which to base customized environment. This document will focus largely on the steps required to take this starting point environment to the next level.

Template Functionality

The VM templates contain functionality that can be accessed directly after the VM has been initialized. The functionality that is available within the template is dependent upon the application in question. This functionality is often limited by the contents of the demo database template. However, the file

[2] All Oracle VMs require networking information at initialization. This is a choice of static versus dynamic host configuration and any ensuing details such as DNS, Gateway etc.

system installation that is delivered with the PIA-AppBatch template – also known as the mid-tier, is complete and binary-compatible with what is obtained by a conventional bare-metal installation of PeopleSoft and its dependent third party products.

The functionality contained within the template is made accessible by the configuration settings that come embedded in the template. This includes the following:

1. PIA domain installation and connectivity to access the Application Server
2. Application Server and Process Scheduler domains
3. Database connectivity – tnsnames.ora
4. Report node definition
5. Integration Gateway registration

The existence of the above configuration settings makes the functionality immediately available following VM initialization. Without these settings the environment would not be available for logging in. These settings would normally be applied by an administrator in a manual fashion but come pre-configured with the VM template.

Deployment Architecture and Resource Requirements

The demo template distribution described in this document is comprised of two separate virtual machine templates. This means that for a complete runnable environment it is necessary to start two separate virtual machines. Each virtual machine is created and initialized independent of the other[3]. Although there are two VMs in this distribution, each VM can participate in a hybrid environment independent of the other template. For example, a VM created from the PIA-AppBatch template, can be used to connect to a pre-existing bare-metal database.

The following table is an overview of the requirements of the two VMs required to run a demo environment:

[3] Although the templates can be initialized independently of one another it is recommended that the database be initialized first. This is because the AppServer and Process Scheduler domains that are started in the PIA-AppBatch templates need to connect to a running database.

TABLE 3. TEMPLATE RESOURCE REQUIREMENTS

VM NAME	CONTENTS	RUNTIME REQUIREMENTS	INITIALIZATION DETAILS	RUNTIME CHARACTERIZATON
PeopleSoft Demo Database	Oracle DBMS 11g PeopleSoft Application Demo DB	80GB Disk[4] 2 GB RAM 2 Virtual CPUs	Network information is provided and the database listener and instance are started immediately.	Demo Database running
PeopleSoft PIA-AppBatch	PeopleTools PIA PeopleSoft Application cd installation Oracle Weblogic Oracle Tuxedo Microfocus COBOL	15GB Disk 2 GB RAM 2 Virtual CPUs	Network stack and database hostname. Defaults can be chosen for all other initialization parameters. The database VM above must be started and available before this VM can be fully initialized.	Application Server domain Process Scheduler PIA Report nodes and integrations configured

VM Deployment

Deploying a PeopleSoft VM environment for the first time is accomplished through the following high level steps:

1. Create an Oracle VM environment with a server pool meeting the hardware requirements provided in the above table
2. Download PeopleSoft VM Templates from Oracle Software Delivery Cloud
3. Import PeopleSoft VM Templates into Oracle VM Manager
4. Deploy the Database VM followed by the PIA-AppBatch VM

[4] The diskspace varies greatly between Applications. HCM and FSCM require ~85GB, Portal Solutions requires 28GB.

5. Sign on to PIA

The steps outlined above refer to the deployment of a demo environment. Steps 1-3 are common to all Oracle VM templates. During step 4 the VMs are initialized. While the VMs are being initialized the administrator submits a series of responses to interactive questions posed by VM that is being initialized. These questions are described below:

Database VM Initialization

The following information is requested during the initialization of the Database VM.

- **Relinking of binaries**: confirmation that it is unnecessary to *relink* the binaries
- **HTTP Port**: the HTTP port that will be used for Oracle Application Express
- **Database listener port**: the port that will be used for the database listener
- **Password**: the password to be used for database accounts
- **DB Startup**: if you want Oracle Database to be started on boot
- **Binary Path**: the full path of the local bin directory
- **SID Name**: the name of the database SID

PIA-AppBatch VM Initialization

The following information is requested during the initialization of the PIA-AppBatch VM:

- **Network Stack**: this is required for all Oracle VM templates when started for the first time
- **Whether you wish to run PeopleSoft Application Server**: if you choose this option Application Server and Process Scheduler domains will be created
- **Database Host**: if you elect to run PeopleSoft Application Server you will be requested to provide the name of the host running the database. The database VM will have been created in the previous step.
- **Install and Compile COBOL**: if the functionality required by the runtime uses COBOL, the option to install and compile COBOL is prompted.
- **Optimize the demo environment**: towards the end of the demo environment initialization the user is prompted for the option to do additional setup. The additional setup is typically achieved by doing a number of SQL inserts for PeopleSoft settings that live in the database. This helps the VM become available out-of-box by defining the integration gateway and the report repository.

Software Packages

Each PIA-AppBatch template contains the following installed software packages:

TABLE 4. INSTALLED SOFTWARE PACKAGES

SOFTWARE PACKAGE	FILE SYSTEM LOCATION	PURPOSE
PeopleTools Application Server	/opt/oracle/psft/pt/tools	Business logic and page processing
Oracle Tuxedo	/opt/oracle/psft/pt/bea/tuxedo	Messaging and lifecycle framework for AppServer and Process Scheduler.
Oracle Weblogic	/opt/oracle/psft/pt/bea/wlserver_10.3	Web Server that hosts PIA
Oracle Database client	/opt/oracle/psft/pt/oracle-client	Used for Application Server and Process Scheduler to connect to database.
Microfocus COBOL Compiler and Runtime[5]	/opt/oracle/psft/pt/cobol	Only included with Applications that run COBOL programs such as Financials or Human Capital Management.
Config Home (PS_CFG_HOME)	/home/psadm2/psft/pt/<peopletools version>	Location of Application Server, Process Scheduler domains and Verity search indexes.

The software packages in the above table depend on additional software. These additional software packages are system packages and are typically delivered independent of the applications above. These software packages include:

TABLE 5. SYSTEM LEVEL SOFTWARE PACKAGES

SOFTWARE PACKAGE	PURPOSE
ncompress unzip	Required by installation programs such as the PIA installation program delivered with the template.

[5] Microfocus COBOL does not come pre-installed in the VM template. After accepting the license agreement terms the Microfocus installation proceeds. You may decline to install Microfocus COBOL without compromising the ability to work with the VM.

libstdc++ 64bit libstdc++ i386	Runtime library for the C++ based software programs in the PeopleTools Application Server and Process Scheduler domains.

The versions of the software packages above will vary depending upon the PeopleTools release that is being installed. This is because PeopleTools will have been developed and compiled on an OS with a specific set of software packages. Failure to install the correct patches may result in unpredictable behavior. In addition, before receiving assistance from Oracle Global Support it is necessary to install the correct versions of the required packages onto the destination host. Consult the certifications details for PeopleSoft on MyOracle Support (MOS) for more information about which versions of software packages are required.

Virtual Machine Configuration

In addition to the software packages and runtime components described above the PeopleSoft VM templates come pre-packaged with a specific configuration. This configuration determines the runtime profile and the way in which it will function. This includes the following:

Users and groups: three primary users are created with the PeopleSoft VM templates. In particular, these are the root user and the PeopleSoft application users **psadm1** and **psadm2**. The **psadm1** user is the owner of the PeopleSoft binaries in the template. The **psadm2** user is the runtime user under whom the application components such as PIA, AppServer and Process Scheduler will be run. Using different users for installation and runtime administration follows the Secure and Decoupled PS_HOME principle first introduced in PeopleTools 8.50.

Kernel Parameters: The PeopleSoft runtime components require specific system level resources to be allocated in order to function reliably. The default kernel configuration values shipped with Oracle Linux (OL) are inadequate for this purpose. This means that the kernel parameters in the shipped templates are revised. These are found in /etc/sysctl.conf

VM initialization: During the initialization of the VM the administrator is presented with a number of questions regarding the runtime functionality of the VM. The responses provided to these questions drive the way in which the VM is initialized and the runtime components that are started. The script that drives this initialization is /opt/oracle/psft/vm/oraclevm-template.sh. This script is only run the first time that the VM is started.

Constructing PeopleSoft VM Templates

This chapter describes the process by which Oracle VM Templates are created. Although this chapter helps understand how to create a virtual PeopleSoft environment it would be necessary to modify the approach in order for the solution to work on other virtualization platforms.

Conventional Template Construction

PeopleSoft follows a methodology that is specific to templates that will run in Oracle VM. Among other things this involves the use of a JeOS toolkit (described later). The JeOS toolkit is necessary because it uses a modified OS image that lends itself best to Xen-based hypervisors. However, it is also possible to build templates for other virtualization platforms without the use of a JeOS toolkit. The process by which this is achieved is described in this section.

Manual Activities Required to Create a VM

If creating a VM template from scratch the steps below must be followed. These steps have been made as generic as possible with a view to being applicable to virtualization platforms other than Oracle VM.

1. Install Operating System and Patches

The Operating System of choice for developing and testing PeopleSoft applications is Oracle Linux. This platform is used in both virtual and bare-metal environments. When building templates for other platforms, keep in mind that the process described here has only been defined for Oracle Linux. Owing to the significant differences between the operating system characteristics, particular attention should be considered when building virtual appliances for Microsoft Windows.

In addition to the installation of the operating system a number of OS patches are typically applied. These OS patches are crucial to the correct functioning of the runtime environment. Patches are normally installed with the help of the OS package manager – RPM in the case of Oracle Linux.

Virtualization platforms have different techniques by which they install OSes into virtual appliances. Some virtualization platforms such as Oracle VM VirtualBox use ISO files. Others require a hardware instance to be converted to a virtual instance – a process known as Physical-to-Virtual (P2V). The mechanism by which this achieved is vendor specific. If using a virtualization platform other than Oracle VM it is suggested that the documentation for that platform is consulted.

2. Install OS Packages

In addition to the operating system a number of packages must be installed. These packages are required by the PeopleSoft runtime. This includes packages that permit the PeopleSoft software to interact with the OS. *Note:* the term package means something very specific to Linux administrators. The concept of a package manager exists on other operating systems but is frequently managed in a very different way. Red Hat based Linux platforms typically use a package manager such as RPM and obtain packages from a Yum database. Package managers vary from platform to platform. The package management framework applicable to the target platform should be used. If using an operating system other than Oracle Linux it is necessary to identify the equivalent software artifacts that need to be installed on your operating system. For example, the C++ runtime equivalent for

Windows is known as the Microsoft Visual Studio C++ Runtime Redistributables. This is a low level software product that allows C++ programs to be executed on the host operating system. The packages required to run PeopleSoft on Oracle Linux are described in the Software Packages section earlier in this document.

3. Install PeopleTools plus Patch

PeopleTools is installed into the file system in a similar way to the third party products above. In most cases a PeopleTools patch must be installed immediately after installing the base PeopleTools installation. Essential fixes are delivered by PeopleTools patches and must therefore be installed before using PeopleTools. It is necessary also to install PIA on your virtual machine if you seek to run the Web Server. The steps required to install PeopleTools components are described in the install guides. The install guides are available at Oracle Technical Network. In addition installation guides and other information can be found by searching for the product name and release number on My Oracle Support.

4. Install Third Party Application Software

A number of third party software products are required by PeopleTools in order to function. Similar to the PeopleTools installation program these third party products can largely be silently installed. Using a silent installer[6] obviates the need for an interactive installation. There is more information on this process and sample silent response files later in this document. The third party products that must be installed are as follows:

- **Tuxedo plus patch:** Installed before running Application Server / Process Scheduler domains, Tuxedo is installed outside PS_HOME. The certified Tuxedo patch required by PeopleTools must also be installed.
- **Verity**: Installed into PS_HOME, Verity is required to support Search related functionality in the PeopleSoft application runtime. Note that PeopleTools 8.52 and later can use Oracle Secure Enterprise Search (SES) en lieu of Verity. If SES is preferred there are a series of runtime configuration settings that must be applied to enable Search functionality. These steps are documented in the PeopleTools Installation Guide and Search Technology PeopleBooks.
- **Web Server**: A web server installation must be present before any PIA domains can be deployed. This is because the PIA installation program requires the file system location of the web server to be supplied. This document assumes that Oracle Weblogic is used as the web server platform.

[6] Silent installation is also known as unattended installation.

- **COBOL**: Some PeopleSoft Applications such as Financials and Human Capital Management require a COBOL compiler and runtime to be installed. Because COBOL is licensed separately from PeopleTools it is necessary to obtain a license independently from your COBOL compiler vendor. This is the same requirement as conventional bare-metal environments.

Other products may also be required by your installation. For example, products such as Microsoft Excel or reporting products such as Crystal Reports Enterprise. It is the responsibility of the template creator to identify these requirements and deploy them consistent with the bare-metal installation requirements of the product.

5. Install PeopleSoft Application

PeopleSoft Application objects largely reside in the database. However, COBOL, SQR programs, and other file system objects are part of a PeopleSoft Application. These file system artifacts are added via the installation program. The installation program operates very similarly to the PeopleTools installer. This means that the same type of silent installation is used for PeopleSoft Applications and PeopleTools. The PeopleSoft application is installed into the pre-existing PS_HOME where PeopleTools was installed. Depending upon the PeopleSoft Application, any number of bundles, maintenance packs or feature packs may be required. This is dependent upon the Application in question. Consult the release notes for the Application for more information. The release notes for the Application are available from My Oracle Support.

6. Set Local Environment

Prior to working with your PeopleSoft installation a number of changes must be made to your environment. These include the following:

- **Set kernel parameters**: The PeopleSoft runtime environment depends on a number of kernel parameters to be changed from the defaults delivered with your OS. These primarily concern Inter Process Communication (IPC).
- **Database Connectivity**: The Application Server and Process Scheduler domains database connectivity must be setup and configured. This will be explained in your Database vendor's documentation. The approach is identical to the approach taken in a conventional bare-metal deployment.
- **Set environment variables**: A number of path entries must be set correctly before creating and configuring PeopleSoft domains. For example, the Tuxedo runtime directory - TUXDIR, must be added to the PATH environment variable to ensure that its binary content can be located by PeopleSoft programs.

7. Compile COBOL

As noted above, some products in PeopleSoft applications require COBOL in order to function. Before the programs within a PeopleSoft application can be run the COBOL source code must be compiled and linked. This is documented in the PeopleTools Installation Guides.

8. Create Process Scheduler and Application Server Domains

Before the PeopleSoft environment can be used interactively it is necessary to create and start the mid-tier domains – Application Server and Process Scheduler. The configuration settings applied will reflect the preferences of your organization. Creating the domains is achieved through the PSAdmin command line utility. Configuration settings can be modified interactively through a text editor or by using OS level utilities such as 'sed' to programmatically change the configuration files. Much of this draws on knowledge PeopleSoft System Administrators would already possess and is documented extensively in the System and Server Administration PeopleBooks.

9. Install and Configure PIA

Once the Application Server and Process Scheduler domains are started and available PIA domains can be installed and started. The PIA domain can be installed interactively on the same or different VM hosting the Application Server domain. Weblogic must be present before PIA can be installed. The installation program for PIA closely resembles the installation program for PeopleTools. This is documented in the PeopleTools Installation Guides.

10. PIA Level Configuration Activities

Some configuration setup must be completed after the environment is up and running. These configuration settings are applied by logging in to PIA. These configuration settings are required to set up post-installation settings such as the Integration Gateway, Process Scheduler report nodes etc. These steps are equivalent to those that would be followed in a conventional bare-metal environment.

PeopleSoft Oracle VM Templates – Our Construction Process

The process by which these templates are constructed relies heavily upon the JeOS toolkit and the guidelines provided in the Oracle VM Template Developers Guide. The process by which templates are constructed is specific to Oracle VM. Templates built in this manner will not run unmodified on other hypervisors[7]. Even with modifications they are not guaranteed to be deployable on other hypervisors. This description is provided below to lend insight into the approach taken by Oracle PeopleSoft. This process is not the only way in which to construct VM templates. It is provided here for reference purposes only.

[7] This includes hypervisors built from the open source Xen architecture.

JeOS Toolkit for Oracle VM

The JeOS toolkit is comprised of a minimal Oracle Enterprise Linux OS and accompanying script. The accompanying script known as *modifyjeos* is used to customize this minimal OS. The process of OS modification is what customizes the OS for running specific applications. The JeOS toolkit is capable of building virtual machine templates from an OL JeOS distribution that has been specially created for building customized VM templates. All of the Oracle VM templates made publicly available by PeopleSoft are built on top of the OL JeOS distribution. The modifyjeos utility creates a template by installing application packages into the base JeOS image. In addition to installing software into the JeOS specific runtime characteristics are controlled by the modifyjeos utility. See the related materials section at the beginning of this document for more information. The Oracle VM Template Developer's Guide is useful for manually constructing templates. Oracle also provides a GUI –based utility that allows OVM templates to be constructed via an intuitive user interface. This is known as Oracle VM Template Builder. This is described in the related materials section above.

Manual vs Automated Template Construction

With its rich set of command line switches the modifyjeos utility (noted above) offers strong scripting capabilities. This means that it is possible to create VM templates without manual involvement by writing carefully crafted shell scripts. This is the approach taken internally by Oracle to create the released PeopleSoft VM templates.

Because Oracle creates templates very frequently it is desirable to automate the process by which templates are constructed. This means that when our software is built the Oracle VM templates are constructed automatically without any manual intervention. These templates can be immediately consumed by users, testers and developers.

Although Oracle internally creates templates using an automation process it is also possible to manually create templates. This is probably the most common approach taken by those creating virtual environments. This is a suitable approach when creating a one-off template or developing a template that requires a series of manual steps for which there is no automation alternative. Manual creation of templates is the approach described in the Oracle VM Template Developers Guide. This involves creating an empty VM from the base Oracle Linux JeOS distribution, logging into the VM and installing the required software. After the software has been installed and the runtime configuration has been created the VM can then be *saved as a template*. Saving a VM as a template is a feature that is offered by Oracle VM. However, a VM typically contains configuration settings that bind it to unique resources such as an IP address or hostname. This means that any VM saved as a template must have these unique settings revisited before a new VM can be created from it. The approach by which this can be achieved is described in the later section 'Clone and Copy'.

RPMs

RPMs are essential to the process of automated template construction. This is because the JeOS toolkit described above uses RPMs as the delivery mechanism for software to be installed into the template. In fact, any software to be installed into the template must be packaged as an RPM before modifyjeos can install it into the template. This means that all of the software packages included during template creation are first placed into RPMs as an intermediary step. In particular all of the software that contributes towards a running PeopleSoft system is RPMized before the template is created. Each of the software packages defined in Chapter 3 are therefore placed into RPMs. This includes the packages listed earlier in chapter 3.

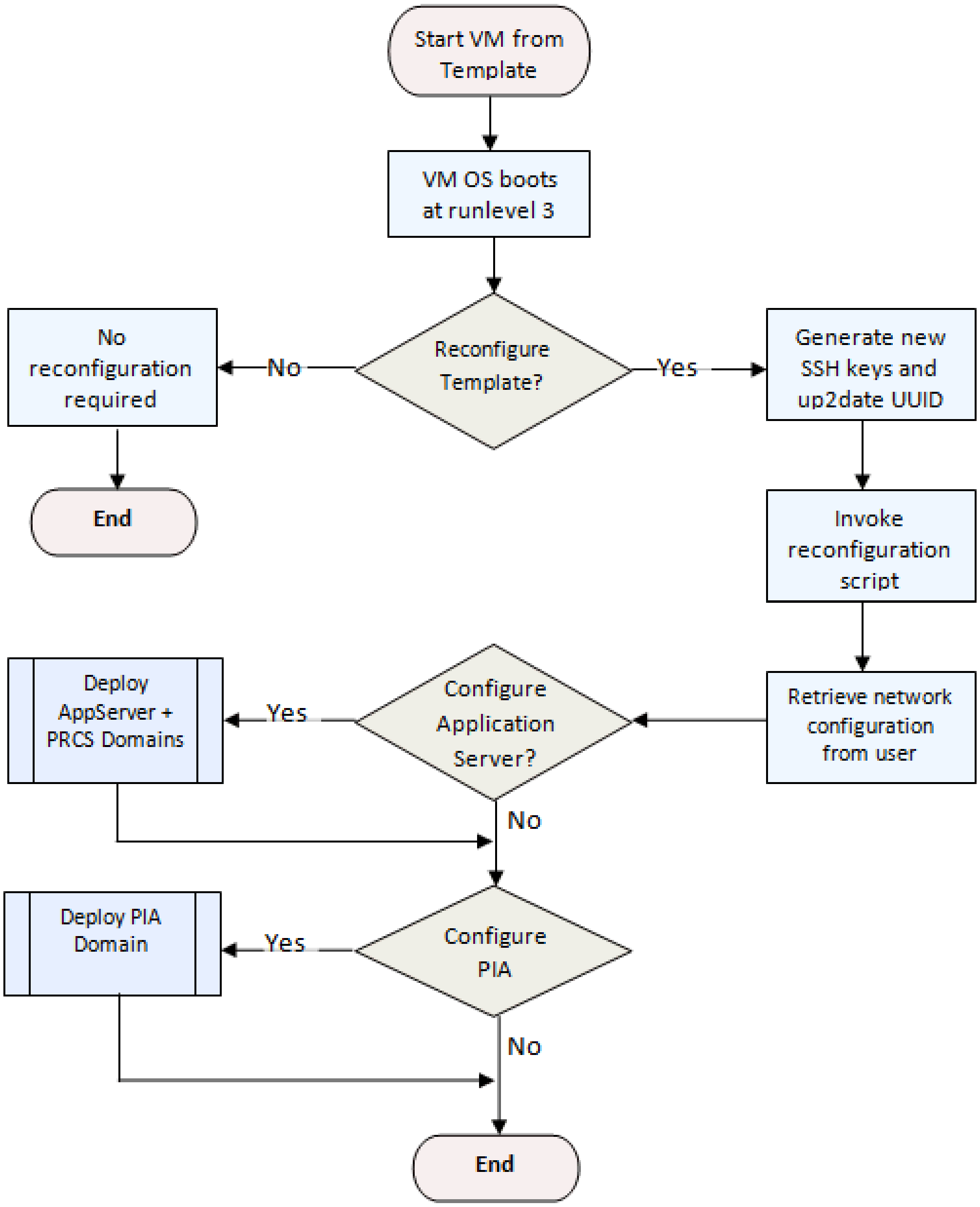
Many of these packages are simply wrappers around the silent install script for the product. This means that the RPM contains directives in its %pre and %post sections of its spec file which place the binaries in the appropriate location on the file system. In the case of the PeopleSoft and all of its dependent sub-products there are silent installation programs available. This involves invoking the same installation program that is typically used for the product and passing a silent response file as a parameter to the installation program.

RPMs offer a level of convenience and reusability that make sense in a complex build environment such as that used at Oracle. The build environment used within the Oracle software development organization requires full automation and iterative quality control. Using RPMs as a delivery mechanism for template construction is therefore desirable for Oracle's internal template development processes. For Oracle customers seeking to build their own templates it is not anticipated that RPMs would be an absolute requirement. The role of RPMs is explained in this section to provide insight into how the templates made available via Oracle Software Delivery Cloud are constructed

VM First Boot Configuration

As noted earlier the VM templates are shipped with a set of instructions that allow it to self-configure the first time the VM is started. This section explains how this process takes place. This document does not provide a detailed description of the Oracle Linux system initialization process. This is general to any Red Hat Enterprise Linux (RHEL) operating system such as Oracle Linux. Consult the resource center at rehat.com for more information.

When the VM is started it enters runlevel 3. As part of initialization the Oracle VM template reconfiguration script is run. Template reconfiguration is simply a mechanism by which the VM that is being instantiated is setup correctly. This template reconfiguration script is located at /etc/rc.d/init.d/oraclevm-template. This script first determines whether the VM has been previously initialized. If this is the first time the VM has been started since being created this script will run to completion. Once the script has completed it exits and the user is ready to log in to the VM. Log messages are written to /var/log/oraclevm-template.log. The diagram below depicts this process.


Start VM from Template
VM OS boots at runlevel 3
Reconfigure Template?
No
No reconfiguration required
End
Yes
Generate new SSH keys and up2date UUID
Invoke reconfiguration script
Retrieve network configuration from user
Configure Application Server?
Yes
Deploy AppServer + PRCS Domains
No
Configure PIA
Yes
Deploy PIA Domain
No
End

The VM reconfiguration script checks to see if the network stack has been configured on the VM. If not, a function is called that prompts the user for the network information such as IP address, hostname, etc that should be used in the VM.

When the basic OS setup has been completed the initialization of the PeopleSoft components commences. This occurs by invoking the PeopleSoft template reconfiguration script /opt/oracle/psft/vm/oraclevm-template.sh.

The primary purpose of this script is to ensure that the software components required to run PeopleSoft are installed and configured correctly. Because many of the configuration settings can only be applied after the host has been configured to run on the network these steps must occur at the end of the VM initialization process. For example, before a Tuxedo domain can be created the hostname must be assigned. The template reconfiguration script implements the steps outlined in the VM Deployment section earlier in this document. The logic in the script is not particular to Oracle VM. These scripts will function correctly on any Red Hat Enterprise Linux system.

Tips for Creating Templates for Other Virtualization Platforms

Other virtualization platforms offer similar mechanisms for packaging operating systems and applications as virtual machines. The terminology varies from vendor to vendor. In addition to being known as VM Templates these are known variably as appliances or images. This section will continue to use the Oracle terminology and refer to them as templates. There are two ways by which templates can be assembled for other virtualization platforms. These are described below.

Conversion of Existing Templates

In its simplest form a template is comprised of one or more block devices and a set of descriptors by which the template is interpreted by the virtualization platform. The block devices are typically portable between virtualization platforms that support the file system type. This is because they are representative of a file system that can be mounted by a POSIX based operating system. This takes some manipulation and requires some understanding of how UNIX file systems are mounted.

A template also will contain one or more configuration files. In the case of Oracle VM 2.x templates this is the vm.cfg file[8]. Such configuration files are typically proprietary. This means that their format is not well understood by other virtualization platform vendors. This typically inhibits unmodified portability of templates between virtualization platforms. Therefore the format of the vm.cfg file that accompanies Oracle VM templates will not be recognized by virtualization platforms other than Xen-based virtualization vendors. Other virtualization platforms will have similar files containing meta-data about the template or virtual appliance. This meta-data contains information such as:

[8] Although the Open Virtualization Format (OVF) presents a platform neutral representation format for such meta data, Oracle VM templates still use the proprietary vm.cfg format. Virtualization vendors are starting to release virtual appliances using OVF. The vm.cfg format can be manually translated to the OVF format if required.

- The file system of the disk images of the template. e.g: ext3 ext4.
- The name of each disk image that is contained in the template.
- Information about the device and how it should be mounted.
- The virtual resources required by each VM when initialized: storage, memory, virtual CPUs etc.
- Other proprietary information about the template that is used specifically by the virtualization platform.

Very often the above concepts are represented in different ways by different vendors. It is however possible to transpose from one vendor's format to another. With intimate understanding of the descriptor fields in the vendor's format it is possible to manually change the template such that it can be run on a different virtualization platform. Although currently challenging and a manual step, this is a process that is expected to become easier with the adoption of the Open Virtualization Format (OVF) as a vendor neutral standard.

Learning how to port templates from one virtualization platform to another is one way by which the existing templates can be exploited. This eliminates the need to assembling the pre-requisite software packages and creating a new template from scratch.

Conversion toolkits are available for virtual templates. Most of these toolkits will convert the configuration file into a platform neutral OVF file. The OVF format is hopefully understood by the virtualization platform on which the template is going to run. Oracle does not provide or certify any tools that will convert the vm.cfg file into the platform neutral OVF format.

Example: Running PeopleSoft OVM Templates in VirtualBox

Oracle VirtualBox is a lightweight solution that permits running virtual appliances on the desktop. This complements the core Oracle VM product that is intended primarily for use in server rooms and multi-host virtualization deployments. It is desirable for demonstration purposes to run virtual appliances on a single machine. This lighter approach permits non-technical users to access virtual appliances and focus on the functional content of the application.

OVM templates cannot currently run unmodified in Oracle VM VirtualBox. The raw disk image format in which the VM templates are delivered cannot be read directly by VirtualBox. It is however possible to use VirtualBox utilities in order to convert the raw disk images to a format that can be used directly by VirtualBox. This section assumes that VirtualBox 4 or later has been installed on the system on which the virtual appliance will be running.

It is recommended that the host on which the VMs are run have sufficient resources to fulfill the hardware requirements stipulated earlier in this document. This is especially important if multiple PeopleSoft VMs will run on the same host. In addition, the network configuration for your VM must be taken into consideration. If your VM needs to be reached from outside it must have an ip address that is distinct from the host OS. For example, this might occur if you are allowing an external PIA connection to access the AppBatch VM. The subject of network configuration is related to how the VM is configured. The VirtualBox User Manual contains information on this topic.

Step 1: Download Oracle VM Templates

Download the OVM templates from the Oracle Software Delivery Cloud for the PeopleSoft Application you desire. Once downloaded these templates need to be extracted. Instructions accompany the downloaded packages which explain how to extract the templates such that you have a series of files as follows:

- System.img: the operating system image
- PSTOOLS.img: the application disk image containing the software packages in Table 4 above.

This is necessary because the templates are typically split into multiple files which need to be recombined following download.

Step 2: Convert to Virtual Box compatible format

The VM disk images noted above must be converted to a format that can be understood by Oracle VM Virtual Box. Specifically the images must be converted to the VDI format. This can be achieved using the VBoxManage utility that is part of the VirtualBox distribution. This is documented extensively in the VirtualBox User Manual. For example, to convert the application disk image above use the following command:

```
VBoxManage convertfromraw  PTOOLS.img  PSTOOLS.vdi
```

Step 3: Create guest OS VM

The OS image contained within the Oracle VM templates is not suitable[9] for use in VirtualBox. This means that a different kernel from that delivered with the VM template is required. For that reason the

[9] VirtualBox requires the Linux OS kernel to be HVM. The modified OS shipped with PeopleSoft OVM templates is PVM.

System.img described in the previous step cannot be used to boot the VM. At this point it is necessary to create your own kernel image. This is achieved by obtaining the installation binaries for Oracle Linux and installing this into a Virtual Machine. The version and bitness of the Oracle Linux distribution must match that of the VM template that you are using.

Once the downloaded installation media - ISO file(s), are available it is possible to mount the installation cd into a new empty VM and run the OS installation program. This is a generic step that can be followed for any new VM that you wish to create from an installation program. This is documented in the VirtualBox User Manual.

Having created the guest OS it is also necessary to install any OS patches and additional packages required by the PeopleSoft runtime. This is documented in the above section - 'Manual Activities Required to Create a VM'. Specifically examine steps 1-2 for hints about what is involved.

Step 4: Attach the disk image to your Virtual Machine

Having created a base OS VM in the previous step it is now possible to attach the application disk image to this new VM. This is a general step to be followed whenever seeking to attach additional disk images to an existing VM. See Storage Settings in the VirtualBox User Manual for more information about attaching additional virtual hard disks to an existing VM.

Step 5: Mount application disk image

Once you have attached the disk image you must start the VM. Having started the VM it is necessary to invoke a number of administrator commands in order to mount the required partition of the application disk image. This is achieved as follows:

1. **Identify disk partition:** the disk will now have been attached to your VM as a *device*. This device will be accessible in /dev. Depending upon which devices have already been allocated this may appear at a different location. For example, this may be at /dev/sdb or /dev/sdc. You can identify where the disk device is located using the shell command **fdisk**. Take note of which partition within the correct device contains the application content.

2. **Connect the disk partition to a loop device:** before mounting the required disk partition it must be connected to a loop device. The partition that you identified using fdisk above must now be connected to a loop device using losetup as shown below:

   ```
   losetup  <loop device> <device partition>
   ```

   For example:

   ```
   losetup  /dev/loop1 /dev/sdb1
   ```

3. **Mount the application file system:** Now that the loop device has been set the next step is to mount that drive position to the required location in the VM file system. It is necessary to mount

the PeopleTools Application image at a very specific location. This is because a PeopleTools installation (PS_HOME) contains a file called psconfig.sh. This file contains hardcoded path locations which require PS_HOME to always reside at the same file system location. This means that it is necessary to mount at the file system location /opt/oracle/psft/pt as follows:

```
mkdir –r /opt/oracle/psft/pt

mount /opt/oracle/psft/pt  <loop device>
```

For example

```
mount /opt/oracle/psft/pt  /dev/loop1
```

If this mount directive should be permanent such that is will be automatically remounted after the system is restarted it is necessary to make an entry in /etc/fstab

4. **Verify the application image subdirectory:** examine the contents of the file system to verify that all file system content is intact. See the below example verification steps:

```
ls –al /opt/oracle/psft/pt

drwxr-xr-x  8 psadm1 oracle  4096 Sep 16 16:51 bea
drwxr-xr-x  3 root   root    4096 Sep 16 17:32 cobol
drwx------  2 root   root   16384 Sep 16 16:26 lost+found
drwxrwxrwx  6 psadm1 root    4096 Sep 16 16:53 oracle-client
drwxr-xr-x 26 root   root    4096 Sep 16 17:37 tools

du /opt/oracle/psft/pt  -hs
7.5G    /opt/oracle/psft/pt
```

Step 6: Application Setup and Configuration

Now that the VM file system has the required content it is necessary to begin configuring your PeopleSoft environment. This is documented in the above section - 'Manual Activities Required to Create a VM'. Specifically examine steps 6-10 to prepare your PeopleSoft environment.

Customizing PeopleSoft OVM Templates

The delivered templates contain a setup that is intended as a starting point for your environment. The templates are designed to permit customization. Customization is achieved in Oracle VM by making changes to a VM and saving that modified VM as a template. The steps by which this is achieved are described below.

Clone and Copy

In VM terminology cloning is the act of making an exact replica of an existing VM. A typical use-case for cloning is security hardening. It is desirable to clone a VM once steps have been taken to harden, tune and customize it. This permits multiple copies of the same VM to be created and added to a running environment. This saves time typically spent on installing and configuring multiple environments. Cloning therefore helps with scalability and adding fault tolerance to a system. This is often a starting point when beginning to experiment with virtual machines as a way to make an operations center more efficient.

Although there are tools by which to clone with most virtualization platforms, there are typically application specific steps required. The steps by which to make a clone usable will be particular to each application. This section focuses on the process of cloning the PIA-AppBatch VM. The general guidelines and principles can however be applied to the database tier. The process of cloning for the database is typically for backup and demo environment creation rather than immediate participation in a runtime environment.

Some of the concepts described below are general in nature. These are considerations to be taken into account when taking any runtime environment and deploying it on a different host. For example, taking a snapshot of an existing VM and seeking to run that snapshot elsewhere would require you to follow the steps below.

Step 1: Customize VM

What are referred to as customizations in this section can be thought of as any change from the delivered VM template. Examples of such are as follows:

- Enablement / disablement of services. e.g: iptables firewall.
- Reconfiguration / addition / removal of PeopleTools domains.
- Addition of file system customizations such as SQR or Cobol
- Additional file system mount points
- Security or user profile changes

Customizations such as those listed above are typically made in a single VM rather than manually repeating these steps across numerous VMs.

Step 2: Reset VM

Some settings in the VM must be removed before the VM can be used as a template. One such setting is the hostname of a VM. Whether the VM uses a static ip or DHCP it is necessary to reset the network stack on the VM before it can be reused. Without clearing the network stack a hostname clash will likely ensue on the network. Therefore any OS settings that bind the cloned VM to a hostname or other network details must be modified before the VM can be used.

The VM templates come with a number of scripts that help make the VM ready to use when it is first started. One such activity is configuring the network stack. These are exposed by the series of prompts that are presented when a VM is first started. There is a complement to the script that sets up the VM and network stack. This complementary script instead clears the network stack and restores it to its uninitialized state. Once restored to its uninitialized state the VM can be reinitialized with a new hostname and network credentials. The network stack of the VM can be reset to its original state by invoking the following commands in sequence from the shell[10]:

- **/usr/sbin/oraclevm-template –cleanup** removes all Application Server and PIA domains, logs, Config Homes etc. This script also directs the cleaning of the OS level configuration such as network stack, ssh host keys etc.
- **/usr/sbin/oraclevm-template –enable** sets a flag within the VM such that next time it is initialized the VM startup script (/etc/init.d) will prompt for network and hostname configuration.

It is important to note that invoking **/usr/sbin/oraclevm-template –cleanup** above will remove the Config Home that has been created in the VM. Any PIA domains in the VM will also be removed. This may be too wide-reaching for your needs. The cleanup script can therefore be customized and modified to the needs of the user. This can be achieved by manipulating the scripts that are delivered with the VM that conduct initialization and cleanup of the VM. More details are available in the Template Reconfiguration Scripts section.

Step 3: Save VM as a template

Once a VM has been reset as described above it may now be saved a template. This is achieved by stopping the VM and using the option in Oracle VM Manager to save the VM as a template. This new template is often referred to as a *Golden Image Template*.

Saving a VM as a template is documented in the Oracle VM User's Guide at download.oracle.com.

Step 4: Create VMs from Customized Template

New VMs can now be created from the customized template. These VMs may be required for different purposes. Examples include:

- Additional VMs to handle increased demand in production environments

[10] The user must be connected through a VNC console and not through a conventional SSH connection. This is because the activity of resetting the VM will clear the network stack and thus make the VM inaccessible from remote connections such as SSH.

- Internal development and testing
- Upgrade rollout - replace existing VMs

The way in which the VM is deployed, stored and customized is completely at the discretion of the administrator. There is no required paradigm that is proscribed by Oracle. The template consumer must identify the ways in which the technology can present most value to the organization.

Step 5: Initialize VM (Reload Domains etc)

Depending upon the type of customizations applied the VM initialization will vary. If the template reconfiguration scripts have not been modified significantly, the prompts made to the user will match those described in the section 'PIA-AppBatch VM Initialization'.

It is necessary to understand the implications of the customizations that have been made and how those relate to the template reconfiguration scripts.

Tuxedo Domains

The name of the host is keyed into the Tuxedo domain configuration file when a domain is created. This means that when you create an AppServer or Process Scheduler domain it becomes *bound* to that hostname. This association is retained in the PSTUXCFG file that is located in your domain directory within your Config Home.

There are a number of ways to address this, all achievable from within the PSAdmin program for administering Application and Process Scheduler domains. The approach taken depends upon the best match for the situation. These options by which this can be achieved are as follows:

- Recreate your domains.
- Reconfigure Existing Domains
- Use the PSAdmin Import Facility
- Replicate Config Home

All of the above options are documented in the System and Server Administration PeopleBooks.

PIA Domains

PIA domains also have a number of properties that are inextricably associated with the host from which your VM was cloned. These properties prevent the PIA domain from being portable. It is therefore necessary to delete the existing PIA domain and create a replacement. This can be done in unattended mode using the silent install response files described later.

Hibernate

If you suspend or pause a PIA-AppBatch VM via OVM Manager the memory and CPU resources dedicated to the VM will be temporarily released and the current state of the VM will be saved to persistent storage. When the VM is unpaused or resumed the resources will be reallocated to the VM and the state of the VM restored. Existing connections to PIA components will, of course expire and the suspension of the VM will proceed without allowing existing transactions to complete. Reconnection to PeopleTools components should take place seamlessly. For example, users will be able to sign back on to PIA without restarting domains.

Change Database

The VM templates come shipped with a demo database for each PeopleSoft application. Instead of using the demo database users frequently wish to connect the AppBatch VM to a pre-existing database. This can be achieved as follows:

1. Stop all existing Application Server and Process Scheduler domains on the VM.
2. Configure Oracle database connectivity by adding an entry in /etc/tnsnames.ora for the new database.
3. Create a new Config Home. This is necessary because a Config Home is explicitly associated with a database instance due to the singularity of the Verity search indexes located at PS_CFG_HOME/data/search.
4. Configure Application Server and Process Scheduler domains to connect to the new database.

Note: the database platform cannot be changed with the VM template that has been downloaded. This is because the PS_HOME is bound to a specific platform – in this case Oracle, at installation time.

Changing File System Locations

The delivered templates contain software deployed to fixed locations. These file system locations may be inconsistent with file system locations preferred by PeopleSoft system administrators. Customers will frequently have their own internal processes and well defined standards to follow. If wishing to deviate from the delivered file system locations a number of changes must be made such that the VM functions in a seamless fashion. After stopping all PeopleSoft components such as Application Server and PIA domains etc. the steps described below should be followed.

Step 1: Create new file system locations

As delivered the VM template contains a number of disk partitions. These disk partitions are at specific locations known as mount points. It is likely that you will wish to change these mount points to accommodate the new file system locations. Examine the output of the ‘mount’ shell command to identify where the file system has been placed. This mount point can be removed and a new mount point created that represents your preferred software installation locations. The /etc/fstab file will also need to be updated to ensure that any new mount points will be established when the VM is restarted.

These activities are general system administrator activities and therefore not documented in detail here.

Step 2- Manually update entries within files

Entries within files must be changed manually to reflect the change in file system locations. It is necessary to open the following files in a text editor such as vi or emacs and update the required entries. The files that must be updated are as follows.

- **/etc/profile**: contains numerous references to file system paths as environment variables
- **/opt/oracle/psft/pt/tools/psconfig.sh**: sets the various binary search path locations
- **/opt/oracle/psft/pt/tools/peopletools.properties**: the installlocation key reflects the location of PS_HOME

The following scripts are only used to configure the VM when it is first started. These scripts conduct the initialization of the PeopleSoft components upon first boot. If the initialization functionality is required after the file system paths are changed, the following files will also need to be updated.

- **/opt/oracle/psft/vm/oraclevm-template-appbatch.sh**: script used to setup the AppServer and Process Scheduler domains
- **/opt/oracle/psft/vm/oraclevm-template-utils.sh**: script containing utility functions for configuring the VM
- **/opt/oracle/psft/vm/installpia.sh**: script that dictates where PIA is installed

Oracle Weblogic contains a file which binds it to the file system path to which it was installed. The file that requires modification is as follows.

- **/appl/bea/wlserver_10.3/common/bin/commEnv.sh**: file to set the common environment variables. Numerous *HOME*s must be updated within this file.

Step 3: Remove and Recreate symbolic links

A number of symbolic links must be removed and recreated. To remove a symbolic link use the ‘rm’ shell command to remove the existing link. Use the ‘ln’ shell command to recreate the link to point to the new directory location. The following are the links that must be removed. Before removing the symbolic links, examine where they currently point in order to determine where they should be placed. The following table provides information about these links.

TABLE 6. SYMBOLIC LINKS

LINK	DEFAULT (ORIGINAL) LOCATION
/opt/oracle/psft/pt/tools/appserv/psadmin	/opt/oracle/psft/pt/tools/bin/psadmin
/opt/oracle/psft/pt/oracle-client/11.1.0.7-64bit/lib/libclntsh.so.10.1 /opt/oracle/psft/pt/oracle-client/11.1.0.7-64bit/lib/libclntsh.so	/opt/oracle/psft/pt/oracle-client/11.1.0.7-64bit/lib/libclntsh.so.11.1

Step 4: Reload existing Application Server and Process Scheduler domains

By default the AppBatch VM initializes an Application Server and Process Scheduler domain. The binary Tuxedo configuration files within these domains are explicitly bound to the Config Home (PS_CFG_HOME) into which they were created and contain references to content in PS_HOME.

The most reliable approach is to discard the existing domains. The cleanest way to do this is to delete the existing Config Home. See table 4 above for the Config Home location. New domains can be created by following the guidelines for creating domains in the System and Server Administration PeopleBooks. This approach has the effect of removing existing domains, their logs and any search indexes that may have been generated.

If the Verity search indexes have been built and the domain configuration contains customizations it may be preferable to simply reload the existing domain configuration in order to refresh the aforementioned Tuxedo binary files. This can be achieved by entering the domain configuration menu in PSAdmin and reloading the domain. Keep in mind that PSAdmin will only update the Tuxedo configuration if the domain .cfg file has an updated timestamp. This can be achieved by using the touch shell command on the main .cfg file to update its time stamp if no real changes are required within the configuration. Finally, the installlocation key in PS_CFG_HOME/peopletools.properties file should be updated to reflect the new PS_HOME location.

Step 5: Recreate the PIA domains

After moving the file system PIA domains can be changed manually in order to allow them to work seamlessly. However, given that PIA domain creation is quick it is preferable to recreate the domain. This is achieved by running the PIA installation program delivered within PS_HOME/setup/PsMpPIAInstall. See table 4 for more information about the location of installed software packages.

If you wish to preserve the existing PIA domain, manually update the following files:

- **setEnv.sh**
- **startPIA.sh**

- **stopPIA.sh**
- **createThreadDump.sh**
- **singleserverStatus.sh**

Step 6: Restart the VM

Linux does not typically require system restarts as frequently as Microsoft Windows. Due to the impact of the changes affected above it is recommended here. This will have the effect of verifying any new file system mount points are successful. In addition, some server processes will be restarted such as the COBOL License Manager.

Reference Information

The content below is provided for reference only. Earlier sections of this document will refer to content below.

Template Reconfiguration Scripts

The VM template comes with a collection of scripts that play an important role in the setup of a VM. These scripts are invoked under two circumstances:

1. When the VM is first started
2. When the VM is being cleaned up / reset.

These scripts perform a number of functions. These functions include:

- Initializing the operating system
- Configuring the network stack
- Conduct Application initialization

The following table provides an introduction to scripts and additional file system content used when configuring and cleaning the VM. For more information consult these scripts and additional content in the directories containing these scripts. In the interest of brevity not all scripts used during VM configuration are listed.

TABLE 7. TEMPLATE RECONFIGURATION SCRIPTS

SCRIPT	DESCRIPTION

/usr/sbin/oraclevm-template	The primary entry point script for VM initialization. This script takes an optional parameter to perform a specific function such as cleanup or network stack enablement. This script conducts setup steps that take place the first time the VM is started. These setup steps are generic in nature and are typically followed for any pre-built Oracle VM template. Having called the generic network initialization functions, the product-specific Application initialization script is invoked. This is described below. Examine the comments in this script for more information about how this script works and the available command line switches that can be supplied.
/usr/lib/oraclevm-template/functions	This script contains a series of utility functions that are used during VM initialization and configuration. The functions herein are largely used by /usr/sbin/oraclevm-template. For example, the ovm_cleanup_os function is used to clean up the network stack, stop services, remove logs, etc when the user invokes /usr/sbin/oraclevm-template --cleanup
/etc/sysconfig/oraclevm-template	Configuration file that sets variables that determine the location of the Application configuration script. The purpose of this file is: • To set the file system location of the Application VM configuration and cleanup scripts. • Retain a flag indicating whether the VM must be configured when it starts.
THE ROWS ABOVE ARE COMMON TO ORACLE VM TEMPLATES FOR PEOPLESOFT AND OTHER ORACLE APPLICATIONS. THE ROWS BELOW DESCRIBE CONTENT SPECIFIC TO PEOPLESOFT TEMPLATES. THE FILE SYSTEM CONTENT BELOW WILL NOT BE FOUND IN TEMPLATES FOR OTHER APPLICATIONS.	
/opt/oracle/psft/vm/oraclevm-template.sh	Entry point for the initialization of the PeopleSoft components and any of its dependencies. When invoked with zero parameters this script examines the file system content of the VM and determines whether PIA, Application Server, Process Scheduler components will be deployed when the VM is first started. This script also will direct the removal and cleanup of all PeopleSoft components if invoked with the '—cleanup' parameter.

/opt/oracle/psft/vm/oraclevm-template-ext.sh

The purpose of this script is to allow template consumers to add custom steps to the VM initialization and cleanup process. This script acts as a rudimentary plug-in model. This is the entry-point for VM customization. Specifically this script is intended to allow additional VM initialization and cleanup steps to be added in addition to the option to disable the PeopleSoft specific VM initialization steps.

This script contains the high level functions:

- ovm_configure_pre() – invoked *before* any of the PeopleSoft functionality during VM initialization.
- ovm_configure_post()– invoked *after* any of the PeopleSoft functionality during VM initialization.
- ovm_cleanup_pre() – invoked *before* any of the PeopleSoft functionality during VM cleanup / reset.
- ovm_cleanup_post() – invoked *after* any of the PeopleSoft functionality during VM cleanup / reset.

The above scripts will be invoked when the VM is being initialized or cleaned up.

Users wishing to suppress the default functionality of the VM initialization can set the OVM_PEOPLETOOLS variable here to FALSE or 0 (zero).

This script and the functions defined above are the mechanism by which the PeopleSoft application configuration is achieved – default settings such as database name, user names and passwords and overridden in a per-Application basis via this script. The VM template initialization process for all PeopleSoft applications is the same except for the contents of this script. This allows the same base PeopleTools scripts to be used despite fundamental differences in runtime requirements of different PeopleSoft applications.

Users are invited to further extend and modify this script in order to customize their VM template. Users are recommended to restrict their changes to this script and your additional scripts that it may call. By limiting the number of delivered files that are modified the burden of uptaking newer versions of the Oracle VM template is reduced.

Because this script is sourced last by /opt/oracle/psft/vm/oraclevm-template.sh, any functions in this script can override functions of the same name in other scripts. This is an additional way in which default functionality can be disabled or overridden.

/opt/oracle/psft/vm/oraclevm-template-env.sh

Sets up report node definitions, integration gateway etc. This script is used primarily for demo environment setup and therefore might need to be skipped or overridden in custom environments.

/opt/oracle/psft/vm/oraclevm-template-appbatch.sh /opt/oracle/psft/vm/oraclevm-template-pia.sh	Conducts the steps required to set-up the PeopleTools AppBatch and PIA components.
/opt/oracle/psft/vm/sql	The VM contains a series of SQL scripts required to set-up PeopleSoft Application demo database content. These SQL scripts are used by the AppBatch VM configuration functions to set-up report nodes, integration gateway etc.
/var/log/oraclevm-template.log	Contains log messages written during the VM initialization process. Log messages are graded as informational, warning and error.

As noted earlier in this document the content above can be changed or manipulated as desired. Many of the functions in the scripts above are intended for reuse. For example, a custom script could be written as follows to reset the OS without removing any of the application components – PeopleSoft domains etc.

```
#!/bin/sh
. /usr/lib/oraclevm-template/functions
ovm_cleanup_os
/usr/sbin/oraclevm-template --enable
```

Silent Product Installation

Software applications are frequently installed using an installation program known commonly as a virtual cd (vcd). This program is conventionally invoked in console mode which means that a series of interactive prompts are displayed. The administrator installing the software provides responses to these prompts. This includes details such as the location to which the software should be installed or other preferences concerning features germane to the application being installed.

To circumvent this interactive installation process a mechanism known as silent installation is frequently supported. This permits installations to be conducted without needing to respond directly to such prompts. This has enormous value to administrators who need to frequently install the same software to multiple targets. Silent installation facilitates scripting the installation process. In order to communicate the preferences otherwise specified interactively, a silent response file can be crafted which directs the installer to implement the selections chosen by the administrator.

Silent response files typically vary from application to software application, many of which are in classic INI format. Examples of these silent response files are provided below.

PeopleTools and Patch

The PeopleTools installation program is invoked in silent mode as follows:

```
setup.sh -i silent -DRES_FILE_PATH=<silent response file> -tempdir <temp>
```

A sample silent response file is shown below

```
USER_INSTALL_DIR=/opt/oracle/psft/pt/tools

DB_TYPE=ORA

USER_INPUT_RESULT_5=false

UNICODE_DB=1

SELECTED_SERVERS=App;Batch;Web;DB

USER_INPUT_RESULT_15=PSEMHUB

USER_INPUT_RESULT_16=80

INSTALL_ERP_CONN=no
```

PeopleSoft Application

The PeopleSoft application follows a very similar format to that supported by PeopleTools. The invocation command is the same as that for the PeopleTools installation program shown above. The same silent response file that is used for PeopleTools can be used for the PeopleSoft Application.

Oracle Tuxedo

Tuxedo can be installed silently as follows:

```
<Tuxedo installation program> -f  <silent install response file> -i silent
```

An example silent response file is as follows:

```
USER_INSTALL_DIR=/opt/oracle/tuxedo

INSTALLER_UI=silent

USER_LOCALE=en

ORACLE_HOME=/opt/oracle

TLISTEN_Port=3050

TLISTEN_PASSWORD=password

CHOSEN_INSTALL_SET=Full
```

The Tuxedo patch must be installed on top of the base installation. The Tuxedo patch is installed by sourcing the tux.env file in TUXDIR and then invoking the installation program that was delivered with the patch distribution. Consult the Tuxedo installation documentation for more information on this subject.

Oracle Weblogic

Oracle Weblogic supports silent mode installation by creating an XML response file. The format of this response file is not conducive to be being displayed here. More information is available in the Weblogic installation documentation at download.oracle.com.

Validation and Feedback

This section documents the real-world validation that this red paper has received. The current version is the first publicly available document on this subject. Feedback will be integrated into this document in subsequent versions.

Customer Validation

Oracle is working with PeopleSoft customers to get feedback and validation on this document. Lessons that are learned from these customer experiences will be posted here.

Field Validation

Oracle Consulting Services has provided feedback and validation on this document. Additional lessons that are learned from field experience will be posted here.

Authors

Mark Johnston – Oracle PeopleTools

Revision History

TABLE 8. REVISION HISTORY

DATE	CHANGE
November 2011	This version is based on Oracle VM Templates created with PeopleTools 8.51.13 and PeopleSoft 9.1 Applications.

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

PeopleSoft OVM Templates - Development
and Customization Guide
December 2011
Author: Mark Johnston

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200

oracle.com

Hardware and Software, Engineered to Work Together