PEOPLESOFT ENTERPRISE

Clustering and High Availability for Enterprise Tools 8.4x – 8.5x

Authors: Sheshi Sankineni
Simon Sy
Hemanth Sundaram
Susan Chen

Last Updated: October 2009

Disclaimer
This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Trademark Information
Oracle, JD Edwards, and PeopleSoft are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

TITLE OF PAPER

10/12/2009

Reference Number:

CRM Product(s):

CRM Release(s):

Revision Number:

Contains:

√

√

√

√

√

√

Table of Contents

Chapter 1 - Introduction

This Red Paper is a practical guide for technical users, installers, system administrators, and programmers who implement, maintain, or develop applications for your PeopleSoft system. In this Red Paper, we discuss guidelines on how to build a fault tolerant PeopleSoft Online Transaction environment, including PeopleSoft Internet Architecture and Portal. The clustering architecture ensures that the deployed PeopleSoft system has no single point of failure and that the system can operate uninterrupted in an event of a HW/SW failure.

Much of the information contained in this document originated within the PeopleSoft Global Support Center and is therefore based on "real-life" systems encountered in the field.

STRUCTURE OF THIS RED PAPER

This Red Paper provides guidance for building Webserver and Application Server clusters.

Keep in mind that PeopleSoft updates this document as needed so that it reflects the most current feedback we receive from the field. Therefore, the structure, headings, content, and length of this document are likely to 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 posted on Customer Connection.

RELATED MATERIALS

This paper is not a general introduction to clustering, fault tolerance or disaster recovery. We assume that our readers shall consult additional reference material for an in-depth understanding of the subject. To take full advantage of the information covered in this document, we recommend that you have a basic understanding of system administration, Internet architecture, network architecture, and PeopleSoft 8 architecture.

This document is not intended to replace the documentation delivered with the PeopleTools 8, 8.14 or 8.4 PeopleBooks. We recommend that before you read this document, you read the PIA related information in the PeopleTools PeopleBooks to ensure that you have a well-rounded understanding of our PIA technology.   Note: Much of the information in this document will eventually get incorporated into subsequent versions of PeopleBooks.

Many of the fundamental concepts related to PIA are discussed in the following PeopleSoft PeopleBooks:

- PeopleSoft Internet Architecture Administration (PeopleTools Administration Tools and PeopleSoft Internet Architecture Administration)
- Application Designer (Development Tools|Application Designer)
- Application Messaging (Integration Tools|Application Messaging)
- PeopleCode (Development Tools|PeopleCode Reference)
- PeopleSoft Installation and Administration
- PeopleSoft Hardware and Software Requirements
- ServerTools (Working with BEA WebLogic Server)

Additionally there is document that discusses administering a multi server weblogic domain configuration. It is HIGHLY recommended that you obtain this document before configuring a multi server WebLogic configuration.

- For PeopleTools 8.44, this document which titled "Enterprise PeopleTools 8.44 and the WebLogic 8.1 Managed Server Architecture" which is available on PeopleSoft's Customer Connection and can be accessed by navigating the following; Customer Connection "www.peoplesoft.com" / Documentation / Documentation Updates / PeopleTools / Server Tools Administration.
- For PeopleTools 8.45, this information will exist in PeopleBooks. Specifically in the "ServerTools" book, as a new chapter titled "WebLogic 8.1 Managed Server Architecture"

Beyond PeopleTools documentation, we recommend that you read the BEA documentation (in HTML format) delivered with the BEA CD-ROM, to gain a thorough understanding of the BEA products that PeopleSoft uses, Tuxedo, Jolt, and WebLogic Server. Refer to your PeopleSoft Installation and Administration book for directions on accessing the delivered BEA documentation

Chapter 2 - The Big Picture

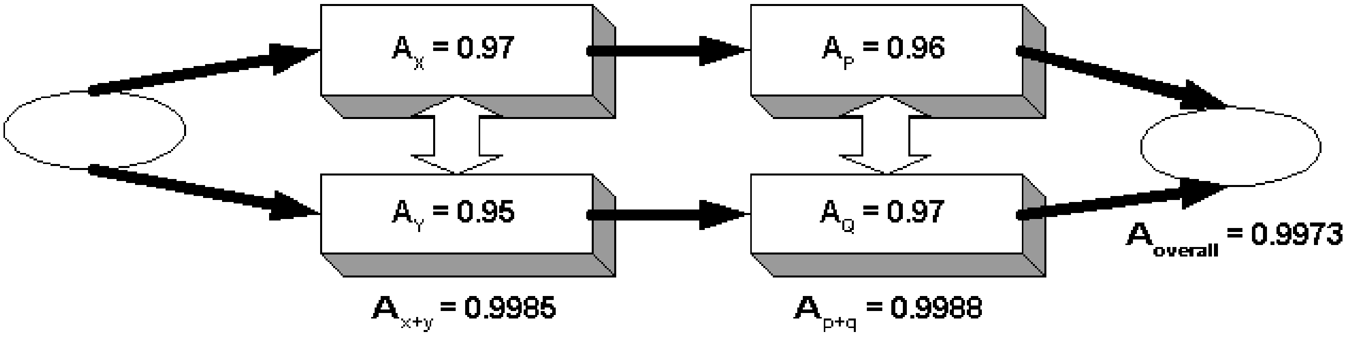
This chapter discusses various components used for scalability and high availability of internet services. Instead of covering all possible configurations/devices, the discussion shall be limited to systems that apply to PeopleSoft architecture and have been tested in the field. The complexity and cost of the system is largely dependent on the required level of Quality Of Service (QOS) of the system. The QOS of a system specifies the level of scalability and fault tolerance the system would provide. In the simplest case there is one server with no guaranteed uptime of service and on the other hand we can build a system to provide 24x7 with better than 99.999% availability i.e. telecommunication grade service. Most of our customers will choose a level of service somewhere in between based on their budget.

Manufacturers of network devices provide MTBF (Mean Time Between Failure) numbers which should be carefully considered. The higher the number the better but it costs more. Do not make a judgment solely based on MTBF without also considering MTTR (Mean Time To Repair) because units that are difficult to repair will eventually contribute to higher down time. The value of MTTR is difficult to calculate because it factors in issues like time to diagnose a problem, availability of parts, engineer's knowledge of the affected unit etc. Calculate availability of overall infrastructure as:

Availability of a component x, A_x = MTBF/(MTBF+MTTR)

Availability of a redundant component group of x and y is $A_{x+y} = 1 – ((1 – A_x) * (1 – A_y))$

Availability of two redundant groups in series to complete a system $A_{overall} = A_{x+y} * A_{p+q}$



The various components to consider in the system are:

Internet Connectivity – For high availability internet connectivity should be obtained from multiple (at least two internet service providers). In the event of a failure of one of the providers users would still be able to access the system via the second provider. The key feature to look for is diversity in connectivity between the two providers, e.g. consider installing leased line for primary provider and satellite or cable modem for the backup. Smaller sites could setup dial backup on backup router, for a more cost effective solution. With cooperation from both the providers it is possible to run full BGP 4 (Border Gateway Protocol) routing protocol for advanced failure detection and failover.

Routers –The router needs to be fault tolerant. At a minimum the network architecture should be dual redundant. The routers could be configured to run in primary/backup mode running either Virtual Router Redundancy Protocol (VRRP) or HSRP (Hot Standby Routing Protocol) for Cisco routers. Under these protocols each unit in a pair sends out packets to see if the other will respond. If the primary fails the backup will take over its functions. Most routers also have certain firewall capability, e.g. packet filtering, port blocking etc. These features should be enabled for added security whenever possible.

Customers using colocation will generally not have access to the router because this is part of the colocation provider’s equipment. In these cases all security features must be implemented within the system using additional equipment (firewall, loadbalancer NAT, reverse proxy server etc).

Switches/VLANS – Switches interconnect all the network devices in a system. To build a redundant system at least two physical switches should be used. In the discussion that follows layer 2 switches are used. Failover for these devices can be configured by using the spanning tree protocol and connecting the devices with a trunk link. The trunk must use redundant interconnect to prevent the LAN from splitting in two. In the configurations shown in this document we have avoided cross connecting switches with routers and hosts. This is a simple configuration that all routers and hosts will support but in an event of a failure of one of the switches half of the servers (all servers connected to the affected unit) in the network are taken offline.

Firewalls – The firewall is possibly the most difficult device to incorporate on a system that is being designed for high availability. In most systems if not properly designed it would soon become the bottleneck. It is not uncommon for extremely high throughput systems to avoid a firewall at the incoming internet entry point. Instead a combination of routers, loadbalancers and reverse proxy servers are used to achieve the necessary level of security for the first tier of the system. High availability with firewalls can be tricky too; most vendors provide some means of clustering capability that allows either an array of identical servers dividing up the load among themselves or an active/active pair of units.

In the following sections we use a 3-pronged firewall. In this device the firewall has 3 interfaces, one for Internet, one for Intranet and one for the DMZ services. This configuration has a single point of protection (security failure) limitation for the Intranet site. If this is not acceptable the 3-pronged firewall should be preceded with another pair of redundant firewalls. It is possible to run loadbalancers to distribute load among identical firewall units

(FWLBS) for greater scalability but the configuration is not simple. To implement the 3-pronged firewall with redundancy it will take 6 extra loadbalancers and 6 extra switches/VLANS to implement.

Loadbalancers – A highly recommended device to achieve high scalability and fault tolerance at a reasonable cost. The current street price for these units range from $5,000 to $50,000. Some units starting at $12,000 can be configured to replace a firewall and provide a hardware SSL accelerator which provides security and scalability at a reasonable cost. Again, a pair should be deployed for redundancy. On most loadbalancers each physical unit can be configured into multiple logical units. Network security and architecture permitting the logical units can be used to loadbalance multiple applications.

Reverse Proxy Servers – Reverse Proxy Servers (RPS) are generally used as part of the security infrastructure. Most sites will deploy them if there is a security concern about IP packets from untrusted users to make it to the production webservers. A RPS provides protection from attacks that are launched to take advantage of vulnerability such as buffer overflow, mal formed packets etc. This also adds another tier to the security architecture. Other sites may use them as a single signon portal server, one which allows RPS authenticated users to access multiple internal systems with varying authentication schemes to be accessed without individual authentication to those systems.

RPS is almost always loadbalanced using a loadbalancer. For PeopleSoft applications a sites domain name mapping will map to the loadbalancer for the RPSs. In this document an example site portal.corp.com should be mapped to a VIP 123.123.123.100 by external DNS systems and this VIP should be mapped to the RPS loadbalancer.

Servers – Servers themselves have a number of fault tolerant mechanism built into them, e.g. redundant network cards, raid array, dual power supply, fault tolerant motherboard etc. As a minimum there should be at least two servers configured as a dual redundant system. Other than the vendor recommended database-clustering PeopleSoft applications do not use any OS provided server-clustering mechanism. This provides greater flexibility for our customers to pick the best of the breed HW/SW solutions.

DNS Servers – A PeopleSoft production system should avoid using DNS name resolution whenever possible. It may be necessary, however, for PeopleSoft Portal or Applications Messaging to be able to access remote servers. If this is a requirement and if adding a /etc/hosts entry for those name(s) is not convenient only then should DNS name resolution from a local server be considered. Under no circumstances should the local DNS servers be allowed to receive DNS updates from remote servers. The local DNS server should also be prevented from sending DNS queries to the remote server for local addresses. So, in other words, the local DNS server should only query the remote server for addresses that are outside the local domain of the site. High availability is maintained by running a primary and a backup

DNS host, connected to two separate switches. All hosts that need access to DNS service should be configured to use a primary and backup DNS host.

Storage – All PeopleSoft data (configuration meta data) and user data is stored in databases. The databases should be stored in some sort of a fault tolerant device e.g. a RAID (Redundant Array of Inexpensive Disks) device. At a minimum the storage subsystem should be chosen to use data striping, e.g. RAID 5 for low cost systems and RAID 10 i.e. 0+1 or 1+0 for high performance systems

Power Supply – A minimum of two UPS (Uninterruptible Power Supply) is recommended. For systems with higher availability requirement the UPS should be backed by power generators and power drop from two separate substations.

Disaster Recovery Plans – Finally all installations small or large must create a disaster recovery plan. For large installations this should include creation of a second data center at a distant geographic location. The current version of the document does not address all aspects of disaster recovery.

VIPs – VIPs are not physical devices. These are IP addresses where the world points its browsers to access the services. These IP address could point to a real webserver in the simplest case. In most of the systems described in this document it will point to a logical service implemented using firewalls, loadbalancers, proxy servers and real servers. A VIP is also the IP address that the sites DNS name shall map to. In this document an example site portal.corp.com is mapped to a VIP 123.123.123.100 by external DNS systems.

REDUNDANT SETUPS

From the components discussed in the previous section we shall configure some common PeopleSoft system layouts. The system layouts will have varying degree of scalability, availability and security. Since each customers site is unique with different requirements it is expected that some parts of a layout may require modification and PeopleSoft Consulting can provide support for that on a case-by-case basis. The basic design assumptions and policies that have been considered are:

Scalability:

- System should be able to scale with demand as much as possible without requiring change of architecture
- Scale with commodity hardware whenever possible
- Scale with the most cost effective solution
- Focus has been mainly to attain highest scalability for the Webserver and Appserver tiers

Availability:

- System should not have any single point of failure in the architecture
- Most single fault shall not reduce system capacity
- Worst case single fault should not reduce capacity by more than 50%
- When multiple options are available for choosing availability, the simpler approach is adopted.
- Active/Passive is selected instead of more complex Active/Active setup of redundancy
- High availability is only restricted to a single site only (for this version of the document), i.e. disaster recovery over two distant geographic location has not been considered

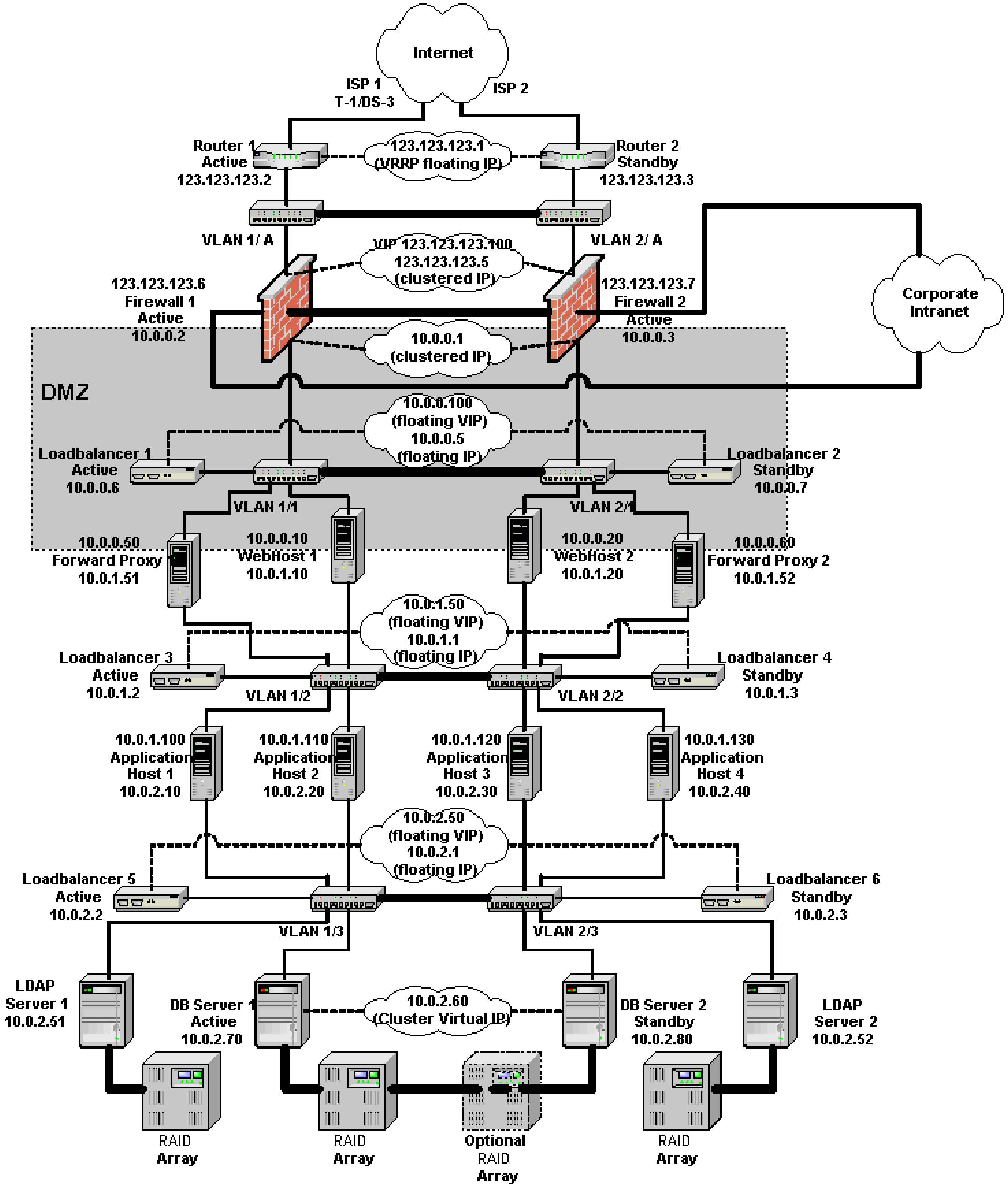
Security:

- System should not have any single point of protection (security failure) in the architecture
- Some security restrictions has reduced the overall scalability of the network
- Name resolution is done via host files instead of using DNS (most cases)
- Static routes are used within the system whenever possible
- PeopleSoft system has been placed on the DMZ network
- There is at least one level of NAT (Network Address Translation) from outside network to the Webserver tier
- The architecture assumes the external/internet as well as internal/intranet network to be untrusted
- The architecture provides at least one extra level of security layer between DMZ and internal network. Should the security of DMZ get compromised the internal network shall still be protected
- Each tier in the PeopleSoft Internet Architecture has been leveraged to provide an additional security tier between the outside network and the protected data
- Portal/App Messaging calls from inside to outside is via a forward proxy
- Default policy of firewall and router is deny all
- We have used a 3-pronged DMZ architecture. This has a single point of protection (security failure) limitation for the Intranet site.

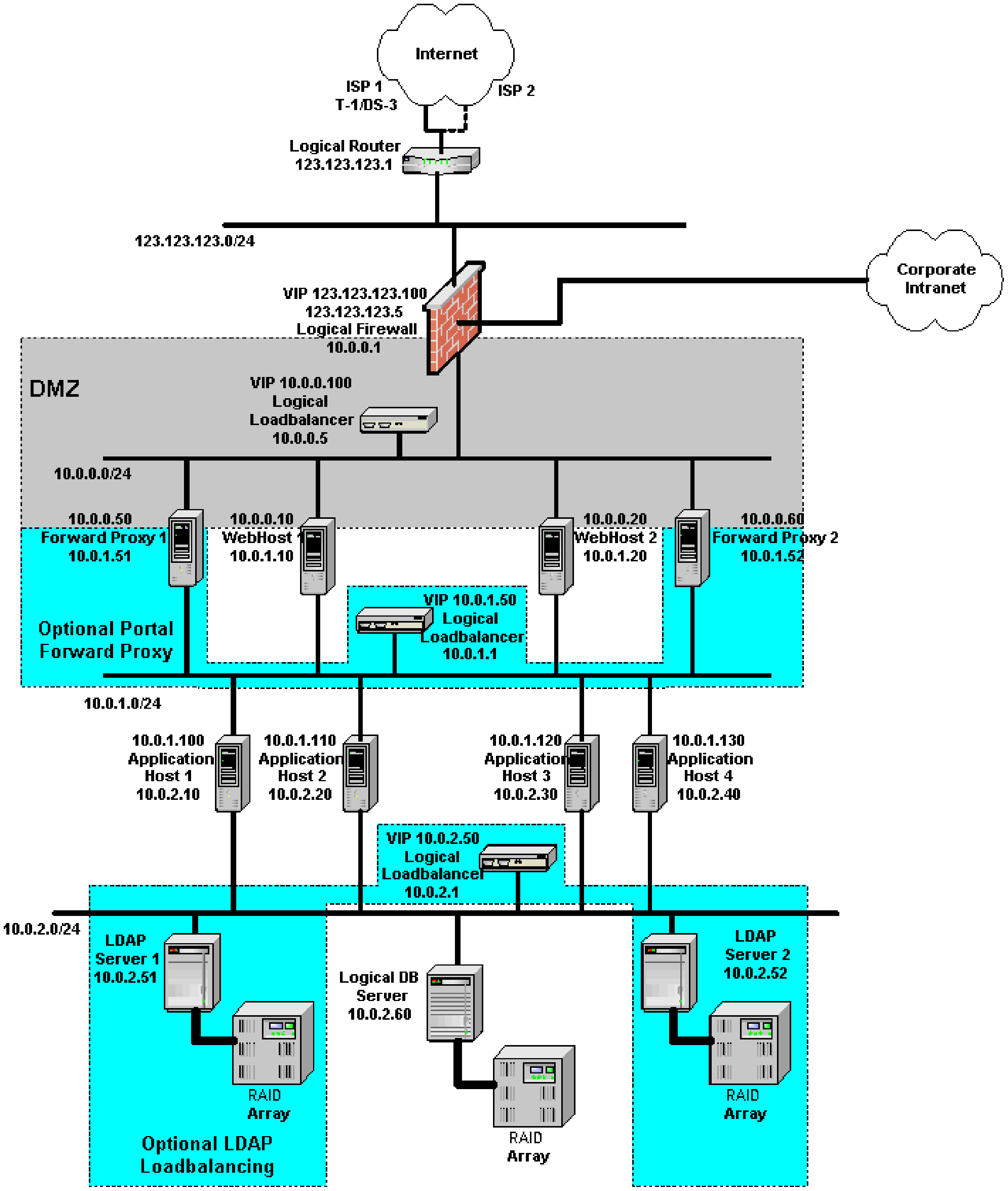
NAT DMZ Redundant Infrastructure

In the NAT (Network Address Translation) DMZ redundant architecture the DMZ occupies a private and non-routable (RFC 1918) Internet address space. The webservers are placed in this private address space in the DMZ. The loadbalancers route packets to the Webservers in the same network. This configuration is only usable if the DMZ is not shared with non-NATable services, such as IPSec and Kerberos. If these services must exist on the DMZ the architecture from the next section must be used.

Physical Layout



Logical Layout



Router Setup

Unit	Router 1 (Active)	Router 2 (Standby)
IP Address	123.123.123.2	123.123.123.3
Subnet Mask	255.255.255.0	255.255.255.0
VRRP IP Address	123.123.123.1	123.123.123.1
VRRP Priority	200	100
Packet filters (only if the available)	Allow only HTTP/HTTPS to PeopleSoft system. If PeopleSoft portal is to call outside allow HTTP/HTTPS to outside from PeopleSoft system. Allow rules as needed by other non-PeopleSoft systems.	Same as Unit 1.

Firewall Setup

Unit	Firewall 1 (Active)	Firewall 2 (Active)
IP Address 1	123.123.123.6	123.123.123.7
Subnet Mask 1	255.255.255.0	255.255.255.0
Shared Address 1	123.123.123.5	123.123.123.5
Default Route 1	123.123.123.1	123.123.123.1
IP Address 2	10.0.0.2	10.0.0.3
Subnet Mask 2	255.255.255.0	255.255.255.0
Shared Address 2	10.0.0.1	10.0.0.1
Default Route 2	None	None
IP Address 3	*	*
Subnet Mask 3	*	*
Shared Address 3	*	*

Default Route 3	None	None

* Based on the Intranet IP address, it can be RFC 1918 address space.

Both firewall units have the same security setup.

Access to PIA/Portal from outside:

Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
HTTP	TCP	Any	80	123.123.123.100	80	Allow
HTTPS	TCP	Any	443	123.123.123.100	443	Allow

Access to Outside from Portal/Application Messaging service:

Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
HTTP	TCP	10.0.0.50	Any	Any	Any	Allow
HTTPS	TCP	10.0.0.50	Any	Any	Any	Allow
HTTP	TCP	10.0.0.60	Any	Any	Any	Allow
HTTPS	TCP	10.0.0.60	Any	Any	Any	Allow

Access to providers DNS server from local DNS server:

Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
DNS[1]	UDP	Local DNS	Any	Provider's DNS	53	Allow
DNS[1]	TCP	Local DNS	Any	Provider's DNS	53	Allow

[1] Do not allow the reverse path i.e. do not allow Provider's DNS updates to reach Local DNS

Static Address Mapping for Inbound Firewall NAT:

External IP Address	Transport Protocol	External Port	Internal Address	Internal Port
123.123.123.100	TCP	80	10.0.0.100	80

123.123.123.100	TCP	443	10.0.0.100	443

Static Address Mapping for Outbound Firewall Reverse NAT:

Source IP	Transport Protocol	Source Port	Translated IP	Translated Port
10.0.0.50	TCP	Any	123.123.123.50	Any
10.0.0.60	TCP	Any	123.123.123.60	Any

Webserver LoadBalancer Setup

Unit	Loadbalancer 1 (Active)	Loadbalancer 2 (Standby)
IP Address	10.0.0.6	10.0.0.7
Subnet Mask	255.255.255.0	255.255.255.0
Shared Address	10.0.0.5	10.0.0.5
Default Route	10.0.0.1	10.0.0.1
Virtual IP (portal.corp.com)	10.0.0.100	10.0.0.100
HTTP Service Port	80	80
HTTPS Service Port	443	443
HTTP Persistence (sticky)	Loadbalancer Cookie	Loadbalancer Cookie
HTTPS Persistence (sticky)	Loadbalancer SSL Sticky	Loadbalancer SSL Sticky

Webserver Setup

The configuration parameters vary based on Webserver Clustering scheme selected look at Webserver Clustering chapter for details.

Unit	WebHost1:Instance1	WebHost1:Instance2	WebHost2:Instance1	WebHost2:Instance2
IP Address 1	*	*	*	*
Subnet Mask 1	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0

Default Route 1	10.0.0.5	10.0.0.5	10.0.0.5	10.0.0.5
HTTP Port	*	*	*	*
HTTPS Port	*	*	*	*
IP Address 2	10.0.1.10	10.0.1.10	10.0.1.20	10.0.1.20
Subnet Mask 2	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0
Default Route 2[1]	10.0.1.50	10.0.1.50	10.0.1.50	10.0.1.50

* See Webserver Clustering chapter for values

[1] Set to none if Proxy Loadbalancing is no used

Forward Proxy Setup

This is an optional setup for Portal, Application Messaging and BI outbound calls.

Unit	ForwardProxy1	ForwardProxy2
IP Address 1	10.0.0.50	10.0.0.60
Subnet Mask 1	255.255.255.0	255.255.255.0
Default Route 1	10.0.0.1	10.0.0.1
IP Address 2	10.0.1.51	10.0.1.52
Subnet Mask 2	255.255.255.0	255.255.255.0
Default Route 2	10.0.0.50	10.0.0.60
HTTP Port	80	80
HTTPS Port	443	443

Forward Proxy LoadBalancer Setup

This is an optional setup for Portal, Application Messaging and BI outbound calls.

Unit	Loadbalancer 3 (Active)	Loadbalancer 4 (Standby)
IP Address	10.0.1.2	10.0.1.3

Subnet Mask	255.255.255.0	255.255.255.0
Shared Address	10.0.1.1	10.0.1.1
Default Route	None	None
Virtual IP for Proxy Service	10.0.1.50	10.0.1.50
HTTP Service Port	80	80
HTTPS Service Port	443	443
Persistence (sticky)	IP Based	IP Based

Application Server Setup

Unit	AppHost1:Domain1	AppHost1:Domain2	AppHost2:Domain1	AppHost2:Domain2
IP Address 1	10.0.1.100	10.0.1.100	10.0.1.110	10.0.1.110
Subnet Mask 1	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0
Default Route 1	10.0.0.1	10.0.0.1	10.0.0.1	10.0.0.1
JSH Port	9000	9020	9000	9020
IP Address 2	10.0.2.10	10.0.2.10	10.0.2.20	10.0.2.20
Subnet Mask 2	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0
Default Route 1	10.0.0.5	10.0.0.5	10.0.0.5	10.0.0.5
LDAP Host	10.0.2.50	10.0.2.50	10.0.2.50	10.0.2.50
LDAP Port	389	389	389	389
LDAPS Port	636	636	636	636

LDAP LoadBalancer Setup

This is an optional setup for LDAP loadbalancing.

Unit	Loadbalancer 5 (Active)	Loadbalancer 6 (Standby)
IP Address	10.0.2.2	10.0.2.3

Subnet Mask	255.255.255.0	255.255.255.0
Shared Address	10.0.2.1	10.0.2.1
Default Route	None	None
Virtual IP for Proxy Service	10.0.2.50	10.0.2.50
LDAP Service Port	389	389
LDAPS Service Port	636	636
Persistence (sticky)	IP Based	IP Based

Database Server Setup

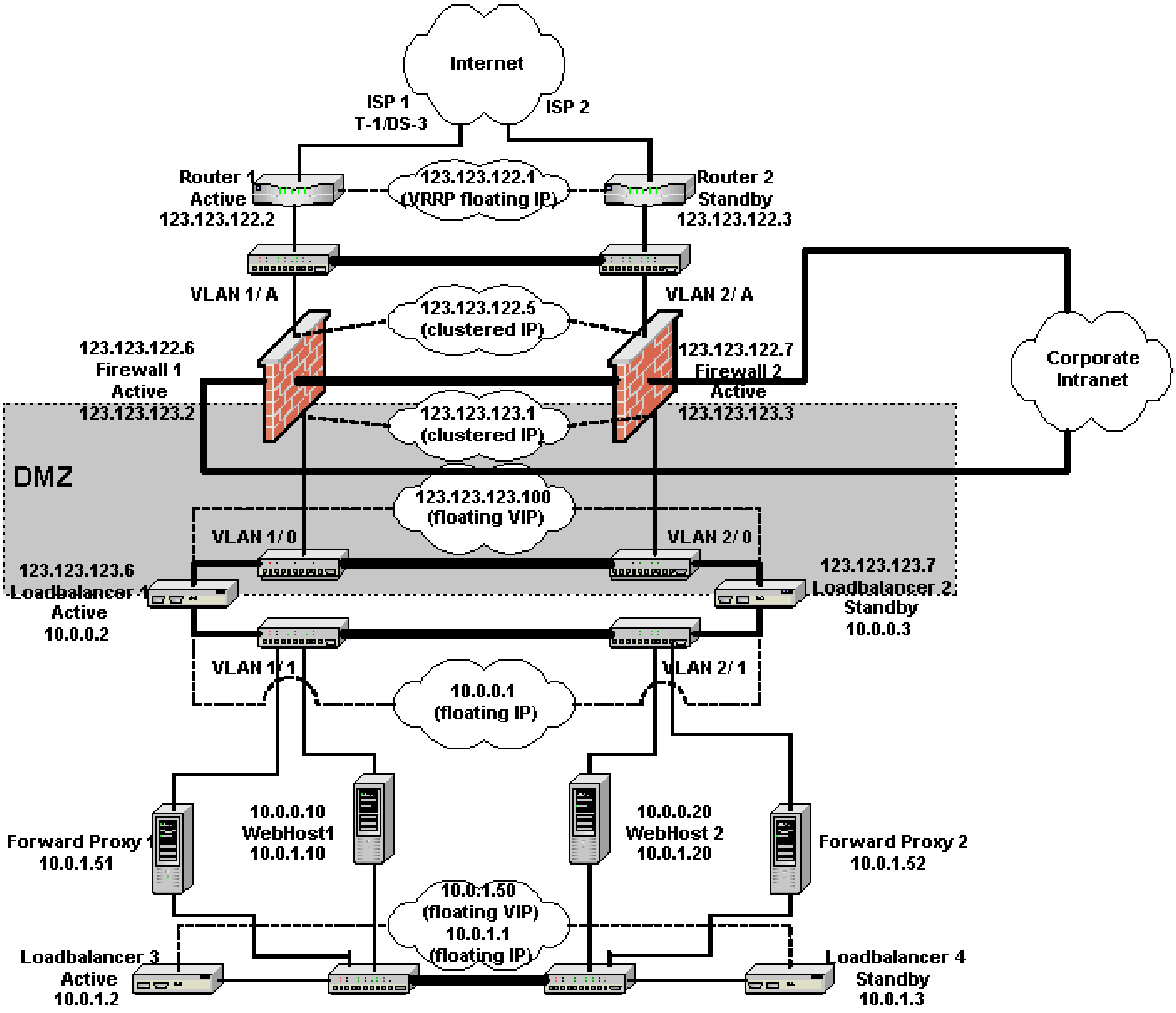
Unit	**DBServer1**	**DBServer2**
IP Address	10.0.2.70	10.0.2.80
Subnet Mask	255.255.255.0	255.255.255.0
Default Route	None	None
Service VIP[1]	10.0.2.60	10.0.2.60
Service Port	DB Vendor Specific	Db Vendor Specific

[1] Required only if Db is clustered

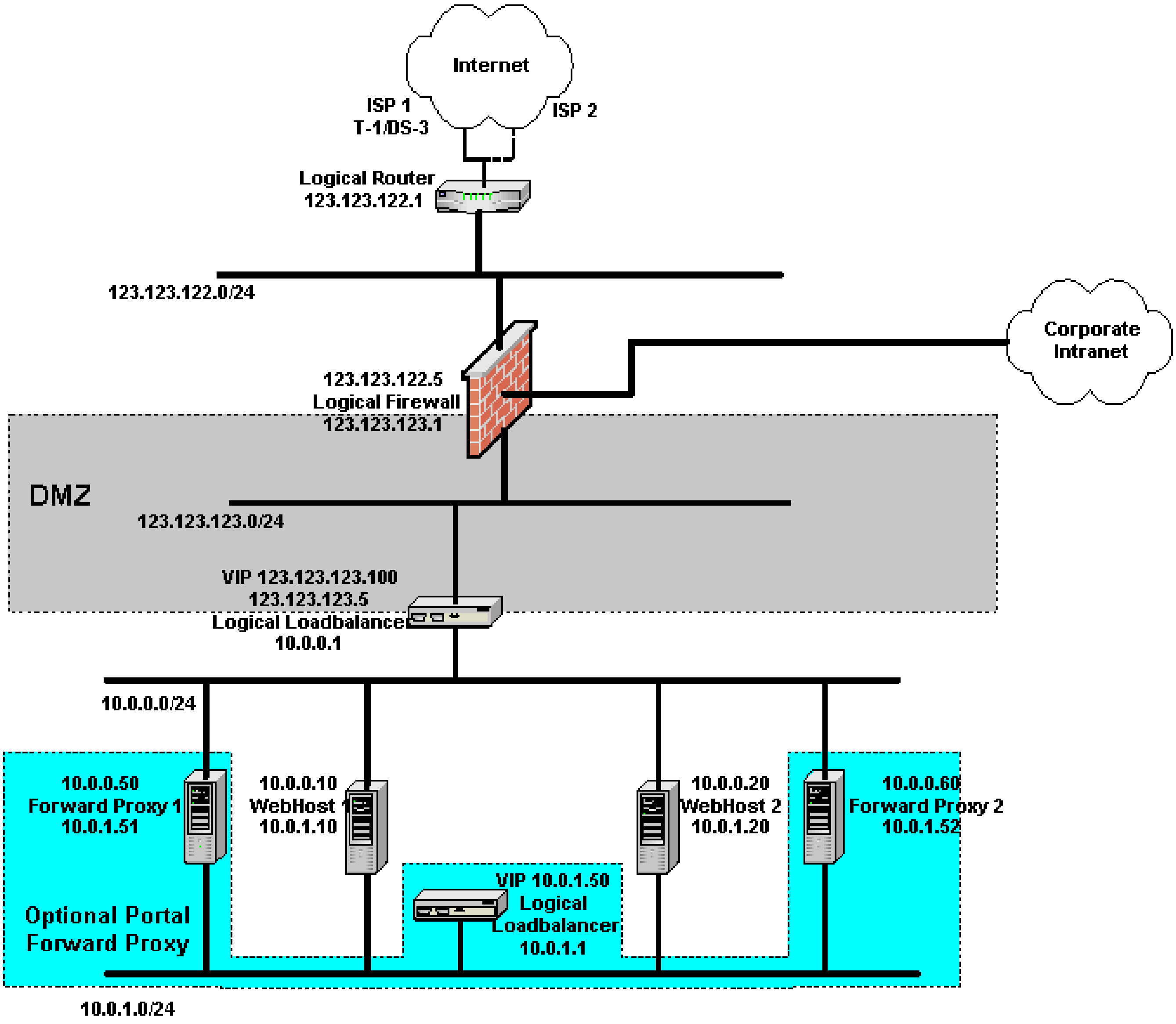
Public Addressed DMZ Redundant Infrastructure

In the Public Addressed DMZ redundant architecture the DMZ occupies a public addressable IP address space. The loadbalancers perform NAT and pass packets to the Webservers which reside in a private and non-routable (RFC 1918) Internet address space. This configuration should be used if the DMZ has to be shared with non-NATable services, such as IPSec and Kerberos. The diagram below only show the modified portion of the architecture as compared to the NAT DMZ architecture and therefore the Application and DB servers have been not been shown.

Physical Layout



Logical Layout



Router Setup

Unit	Router 1 (Active)	Router 2 (Standby)
IP Address	123.123.122.2	123.123.122.3
Subnet Mask	255.255.255.0	255.255.255.0
VRRP IP Address	123.123.122.1	123.123.122.1
VRRP Priority	200	100

Packet filters (only if the available)	Allow only HTTP/HTTPS to PeopleSoft system. If PeopleSoft portal is to call outside allow HTTP/HTTPS to outside from PeopleSoft system. Allow rules as needed by other non-PeopleSoft systems.	Same as Unit 1.

Firewall Setup

Unit	Firewall 1 (Active)	Firewall 2 (Active)
IP Address 1	123.123.122.6	123.123.122.7
Subnet Mask 1	255.255.255.0	255.255.255.0
Shared Address 1	123.123.122.5	123.123.122.5
Default Route 1	123.123.122.1	123.123.122.1
IP Address 2	123.123.123.2	123.123.123.3
Subnet Mask 2	255.255.255.0	255.255.255.0
Shared Address 2	123.123.123.1	123.123.123.1
Default Route 2	None	None
IP Address 3	*	*
Subnet Mask 3	*	*
Shared Address 3	*	*
Default Route 3	None	None

* Based on the Intranet IP address, it can be RFC 1918 address space.

Both firewall units have the same security setup.

Access to PIA/Portal from outside:

Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
HTTP	TCP	Any	80	123.123.123.100	80	Allow

HTTPS	TCP	Any	443	123.123.123.100	443	Allow

Access to Outside from Portal/Application Messaging service:

Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
HTTP	TCP	123.123.123.50	Any	Any	Any	Allow
HTTPS	TCP	123.123.123.50	Any	Any	Any	Allow
HTTP	TCP	123.123.123.60	Any	Any	Any	Allow
HTTPS	TCP	123.123.123.60	Any	Any	Any	Allow

Access to providers DNS server from local DNS server:

Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
DNS[1]	UDP	Local DNS	Any	Provider's DNS	53	Allow
DNS[1]	TCP	Local DNS	Any	Provider's DNS	53	Allow

[1] Do not allow the reverse path i.e. do not allow Provider's DNS updates to reach Local DNS

Webserver LoadBalancer Setup

Unit	Loadbalancer 1 (Active)	Loadbalancer 2 (Standby)
IP Address (VLAN1/0)	123.123.123.6	123.123.123.7
Subnet Mask	255.255.255.0	255.255.255.0
Shared Address	123.123.123.5	123.123.123.5
Default Route	123.123.123.1	123.123.123.1
IP Address (VLAN1/1)	10.0.0.2	10.0.0.3
Subnet Mask	255.255.255.0	255.255.255.0

Shared Address	10.0.0.1	10.0.0.1
Virtual IP (portal.corp.com)	123.123.123.100	123.123.123.100
HTTP Service Port	80	80
HTTPS Service Port	443	443
HTTP Persistence (sticky)	Loadbalancer Cookie	Loadbalancer Cookie
HTTPS Persistence (sticky)	Loadbalancer SSL Sticky	Loadbalancer SSL Sticky

Static Address Mapping for Inbound Loadbalancer NAT:

External IP Address	Transport Protocol	External Port	Internal Address	Internal Port
123.123.123.100	TCP	80	10.0.0.100	80
123.123.123.100	TCP	443	10.0.0.100	443

Static Address Mapping for Outbound Loadbalancer Reverse NAT:

Source IP	Transport Protocol	Source Port	Translated IP	Translated Port
10.0.0.50	TCP	Any	123.123.123.50	Any
10.0.0.60	TCP	Any	123.123.123.60	Any

Webserver Setup

The configuration parameters vary based on Webserver Clustering scheme selected look at Webserver Clustering chapter for details.

Unit	WebHost1:Instance1	WebHost1:Instance2	WebHost2:Instance1	WebHost2:Instance2
IP Address 1	*	*	*	*
Subnet Mask 1	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0
Default Route 1	10.0.0.1	10.0.0.1	10.0.0.1	10.0.0.1
HTTP Port	*	*	*	*

HTTPS Port	*	*	*	*
IP Address 2	10.0.1.10	10.0.1.10	10.0.1.20	10.0.1.20
Subnet Mask 2	255.255.255.0	255.255.255.0	255.255.255.0	255.255.255.0
DefaultRoute 2[1]	10.0.1.50	10.0.1.50	10.0.1.50	10.0.1.50

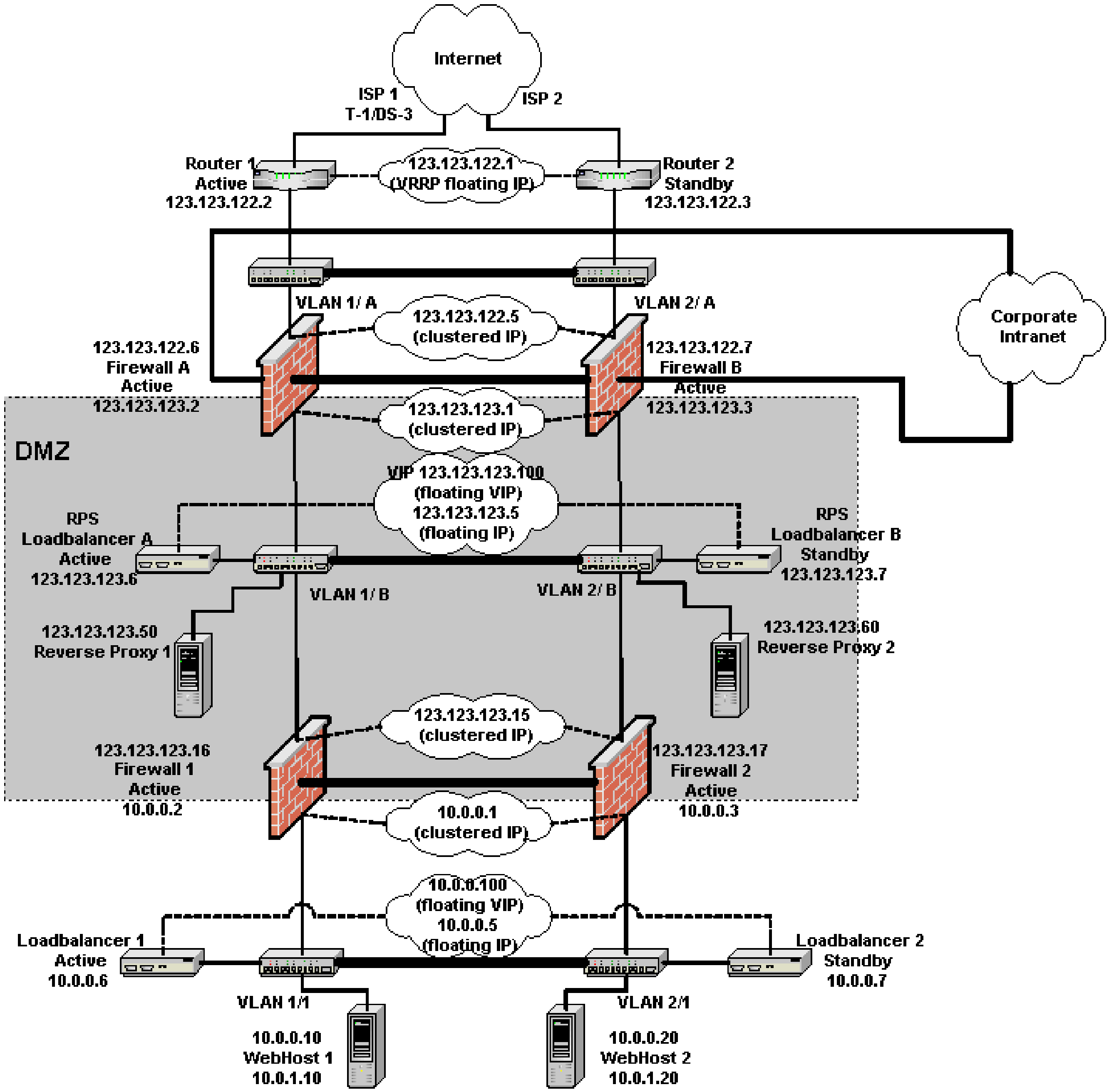
* See Webserver Clustering chapter for values

[1] Set to none if Proxy Loadbalancing is no used

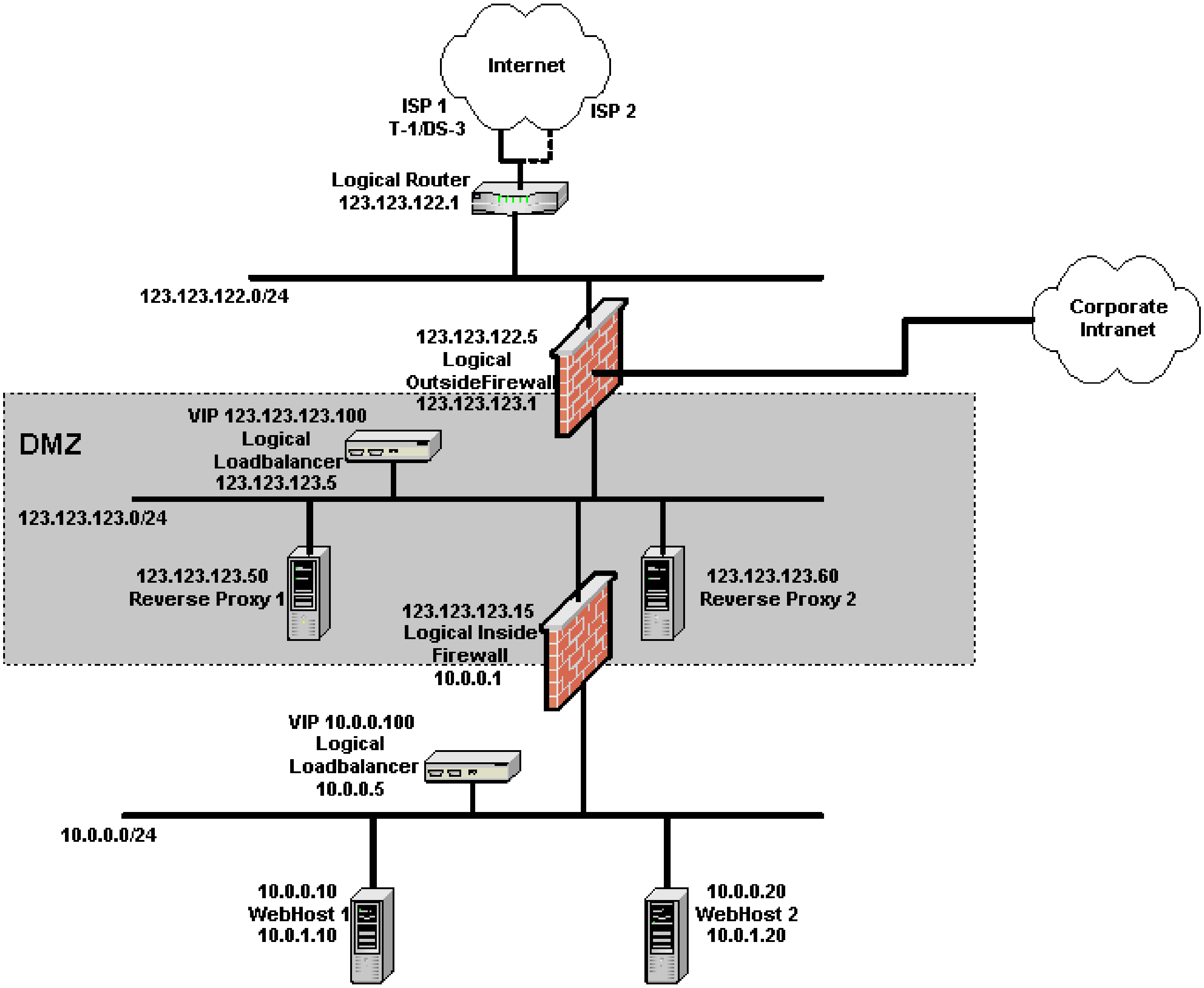
Additional Security DMZ

For a higher security DMZ the following architecture is proposed:

Physical Layout



Logical Layout



Router Setup

Unit	Router 1 (Active)	Router 2 (Standby)
IP Address	123.123.122.2	123.123.122.3
Subnet Mask	255.255.255.0	255.255.255.0
VRRP IP Address	123.123.122.1	123.123.122.1
VRRP Priority	200	100
Packet filters (only if the available)	Allow only HTTP/HTTPS to PeopleSoft system. If PeopleSoft portal is to	Same as Unit 1.

	call outside allow HTTP/HTTPS to outside from PeopleSoft system. Allow rules as needed by other non-PeopleSoft systems.	

Outside Firewall Setup

Unit	Firewall A (Active)	Firewall B (Active)
IP Address 1	123.123.122.6	123.123.122.7
Subnet Mask 1	255.255.255.0	255.255.255.0
Shared Address 1	123.123.122.5	123.123.122.5
Default Route 1	123.123.122.1	123.123.122.1
IP Address 2	123.123.123.2	123.123.123.3
Subnet Mask 2	255.255.255.0	255.255.255.0
Shared Address 2	123.123.123.1	123.123.123.1
Default Route 2	None	None
IP Address 3	*	*
Subnet Mask 3	*	*
Shared Address 3	*	*
Default Route 3	None	None

* Based on the Intranet IP address, it can be RFC 1918 address space.

Both firewall units have the same security setup.

Access to PIA/Portal from outside:

Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
HTTP	TCP	Any	80	123.123.123.100	80	Allow
HTTPS	TCP	Any	443	123.123.123.100	443	Allow

Access to Outside from Portal/Application Messaging service:

Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
HTTP	TCP	123.123.123.50	Any	Any	Any	Allow
HTTPS	TCP	123.123.123.50	Any	Any	Any	Allow
HTTP	TCP	123.123.123.60	Any	Any	Any	Allow
HTTPS	TCP	123.123.123.60	Any	Any	Any	Allow

Access to providers DNS server from local DNS server:

Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
DNS[1]	UDP	Local DNS	Any	Provider's DNS	53	Allow
DNS[1]	TCP	Local DNS	Any	Provider's DNS	53	Allow

[1] Do not allow the reverse path i.e. do not allow Provider's DNS updates to reach Local DNS

Reverse Proxy Server LoadBalancer Setup

Unit	Loadbalancer A (Active)	Loadbalancer B (Standby)
IP Address	123.123.123.6	123.123.123.7
Subnet Mask	255.255.255.0	255.255.255.0
Shared Address	123.123.123.5	123.123.123.5
Default Route	123.123.123.1	123.123.123.1
Virtual IP (portal.corp.com)	123.123.123.100	123.123.123.100
HTTP Service Port	80	80
HTTPS Service Port	443	443
HTTP Persistence (sticky)	Loadbalancer Cookie	Loadbalancer Cookie

HTTPS Persistence (sticky)	Loadbalancer SSL Sticky	Loadbalancer SSL Sticky

Reverse Proxy Server Setup

Unit	RPS1	RPS2
IP Address 1	123.123.123.50	123.123.123.60
Subnet Mask 1	255.255.255.0	255.255.255.0
Default Route 1	123.123.123.5	123.123.123.5
HTTP Port	80	80
HTTPS Port	443	443

Inside Firewall Setup

Unit	Firewall 1 (Active)	Firewall 2 (Active)
IP Address 1	123.123.123.16	123.123.123.17
Subnet Mask 1	255.255.255.0	255.255.255.0
Shared Address 1	123.123.123.15	123.123.123.15
Default Route 1	123.123.123.1	123.123.123.1
IP Address 2	10.0.0.2	10.0.0.3
Subnet Mask 2	255.255.255.0	255.255.255.0
Shared Address 2	10.0.0.1	10.0.0.1
Default Route 2	None	None

Both firewall units have the same security setup.

Access to PIA/Portal from RPS only:

Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
HTTP	TCP	123.123.123.50	80	10.0.0.100	80	Allow

HTTPS	TCP	123.123.123.50	443	10.0.0.100	443	Allow
HTTP	TCP	123.123.123.60	80	10.0.0.100	80	Allow
HTTPS	TCP	123.123.123.60	443	10.0.0.100	443	Allow

Access to Outside from Portal/Application Messaging service:

Protocol	Transport Protocol	Source IP	Source Port	Destination IP	Destination Port	Action
HTTP	TCP	10.0.0.50	Any	Any	Any	Allow
HTTPS	TCP	10.0.0.50	Any	Any	Any	Allow
HTTP	TCP	10.0.0.60	Any	Any	Any	Allow
HTTPS	TCP	10.0.0.60	Any	Any	Any	Allow

Static Address Mapping on Inside Firewall for Inbound NAT:

External IP Address	Transport Protocol	External Port	Internal Address	Internal Port
123.123.123.100	TCP	80	10.0.0.100	80
123.123.123.100	TCP	443	10.0.0.100	443

Static Address Mapping on Inside Firewall for Outbound Reverse NAT:

Source IP	Transport Protocol	Source Port	Translated IP	Translated Port
10.0.0.50	TCP	Any	123.123.123.50	Any
10.0.0.60	TCP	Any	123.123.123.60	Any

Webserver LoadBalancer Setup

Unit	Loadbalancer 1 (Active)	Loadbalancer 2 (Standby)
IP Address	10.0.0.6	10.0.0.7
Subnet Mask	255.255.255.0	255.255.255.0
Shared Address	10.0.0.5	10.0.0.5

Default Route	10.0.0.1	10.0.0.1
Virtual IP (portal.corp.com)	10.0.0.100	10.0.0.100
HTTP Service Port	80	80
HTTPS Service Port	443	443
HTTP Persistence (sticky)	Loadbalancer Cookie	Loadbalancer Cookie
HTTPS Persistence (sticky)	Loadbalancer SSL Sticky	Loadbalancer SSL Sticky

Webserver Setup

All other setup including Webserver setup is the same as NAT DMZ.

Chapter 3 Webserver Clustering

For this version of this document only the PeopleTools 8.4 webserver clustering shall be covered. There are a number of ways to setup a Webserver cluster for a PeopleSoft system. Customers can use Oracle Oracle Application Server or BEA WebLogic or IBM WebSphere as their webserver and use the clustering mechanism provided by the vendor. Please consult the vendor provided documentation for a detailed understanding of Webserver clustering. Note that the architecture proposed here has been adapted for PeopleSoft applications and is therefore somewhat different than the documents provided by BEA or IBM.

BEA WebLogic provides a mechanism to cluster WebLogic webservers and access the cluster via WebLogic proxies. These proxies can be another WebLogic server, Microsoft IIS, Netscape iPlanet or Apache webserver and shall be collectively called “WebLogic proxy” in this document. Please carefully note the distinction between “WebLogic clustering” and “webserver clustering” used extensively in this document. Following up on this distinction some deployment setups will require WebLogic clustering while for others will require WebSphere clustering. When WebLogic clustering is not used a suitable third party loadbalancer must be used. Architecture of these various scheme along with their respective advantages and disadvantages are discussed later in this chapter.

IBM WebSphere provides a mechanism to cluster WebSphere webservers and access the cluster via an HTTP (proxy) plugin. This WebSphere plug-in is installed on an HTTP (proxy) server. Please note carefully the distinction between “WebSphere clustering” and “webserver clustering” used extensively in this document. Following up on this distinction some deployment setup will require WebSphere clustering while others will require WebLogic clustering. When WebSphere clustering is not used a suitable third party loadbalancer must be used. Architecture of these various scheme along with their respective advantages and disadvantages are discussed later in this chapter.

Webserver clustering shall provide scalability and stateless failover. Under stateless failover, if a server fails user sessions shall be restarted with another functioning server. All unsaved data on the failed server shall be lost but the user login shall be preserved. WebLogic clustering requires a cluster license to operate. It should be noted that the WebLogic licenses delivered with a PeopleSoft installation already includes the required WebLogic Cluster license.

WebLogic/WebSphere clustering requires that the cluster is accessed via a WebLogic/WebSphere proxy. This introduces another tier to the data flow and is not recommended for applications with very high scalability requirement.

WEBLOGIC CLUSTER

This section only provides specific instructions of how to setup a WebLogic cluster for a PeopleSoft system. Please consult WebLogic Administrators Guide for a detailed understanding of WebLogic clustering.

PT8.40-8.43: WebLogic Server 6.1
http://edocs.bea.com/wls/docs61/adminguide/index.html

PT8.44 - 8.48: WebLogic Server 8.1:
http://edocs.bea.com/wls/docs81/admin.html

PT8.49: WebLogic Server 9.2:

http://edocs.bea.com/wls/docs92/admin.html

PT8.50: WebLogic Server 10.3.1 (Also known as Oracle Fusion Middleware 11g Release 1):

http://download.oracle.com/docs/cd/E12839_01/wls.htm

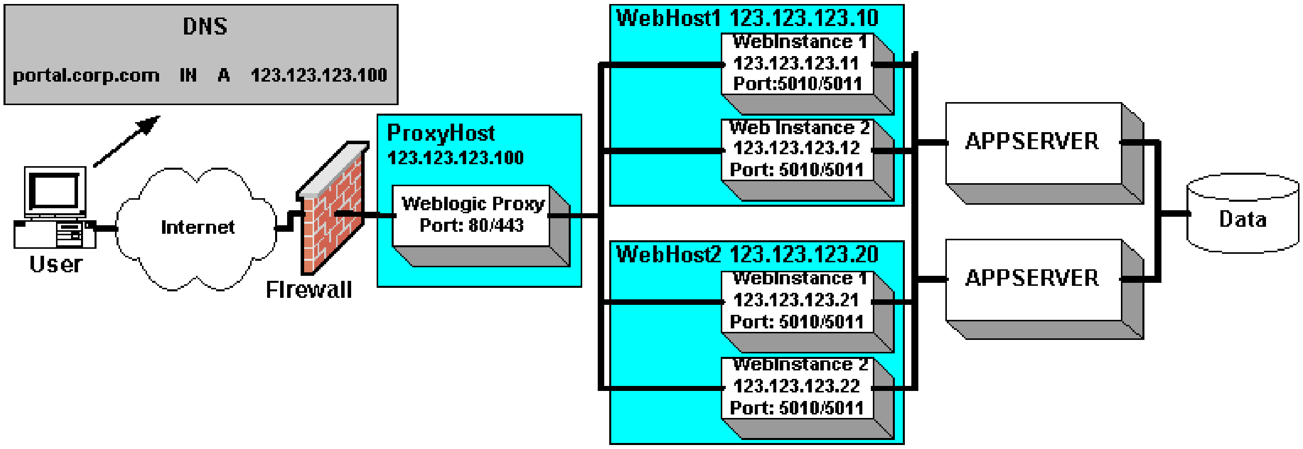
Note that the architecture proposed here has been adapted for PeopleSoft applications and is therefore somewhat different than the documents at the BEA site, but in PeopleTools 8.44 it is all based on using an admin server for administering the WebLogic domain and multiple managed servers serving the PIA applications. When you install PIA on WebLogic 8.1, WebLogic 9.2, and WebLogic 10.3.1 select the option to create a multi server domain.

In the following subsections we discuss two clustering architectures. The first, Simple WebLogic Cluster, has been mentioned here because some systems have been deployed with it. It is the simplest to setup but is also the least scalable and is only applicable to small sites. The second, Advanced WebLogic Cluster, is more scalable and is recommended for customers with high scalability requirement. However, for the most scalable and flexible architecture the Generic Webserver Cluster described in a following section is recommended.

To make the changes described below to your server configuration, such as IP addresses and listening ports use the WebLogic Server console. To access the console start the WebLogicAdmin server by running the startWebLogic script in your WebLogic domain. That script was created when you installed PIA and selected the option to create a multi server domain. The default URL to access the console is http://webserver:9999/console. The ID and password to use is the WebLogic ID/password you specified during the PIA install. Only the “WebLogicAdmin” server will have a console. Managed servers do not have a console but are administered remotely via the WebLogicAdmin server. To create additional servers via the console, such as PIA3, PIA4 as described below you can either clone from an existing server or define a new one.

Simple WebLogic Cluster

In a simple WebLogic cluster there are one or more WebLogic instances per webserver host and there are more than one webserver host for redundancy. There is only one WebLogic proxy server running on a separate host. The architecture diagram is shown below:



For this setup:

- Use WebLogic Proxy server serving HttpClusterServlet web application.
- All instances must use the same HTTP/HTTPS port e.g. 5010/5011 in the example. This requires IP Aliasing if the host does not have dedicated network interfaces for each webserver instance
- One proxy server load balances across all the WebLogic instances
- The proxy must pass all calls to the webserver instances
- WebLogic proxy server and WebLogic servers providing PIA are run as WebLogic Managed Servers

Advantages:

- Simple to setup
- Low cost solution

Disadvantages:

- Proxy adds an additional layer
- WebLogic proxy requires IP Aliasing for running multiple instances
- WebLogic proxy is a single point of failure.

- Very low scalability. The WebLogic proxy will soon become a scalability bottleneck as new webservers are added to the pool particularly when going over SSL.

Setup

The setup is considerably different from the "Big Picture" chapter because the load balancer is missing. To make it easier to understand the architecture illustration shows the mapping for a system where the firewall is not performing any NAT. It is recommended that NAT be performed for greater security.

Proxy Setup

Unit	**ProxyHost** **WebLogic instance name = RPS**
IP Address	123.123.123.100
Subnet Mask	255.255.255.0
Default Route	123.123.123.1
HTTP Port	80
HTTPS Port	443

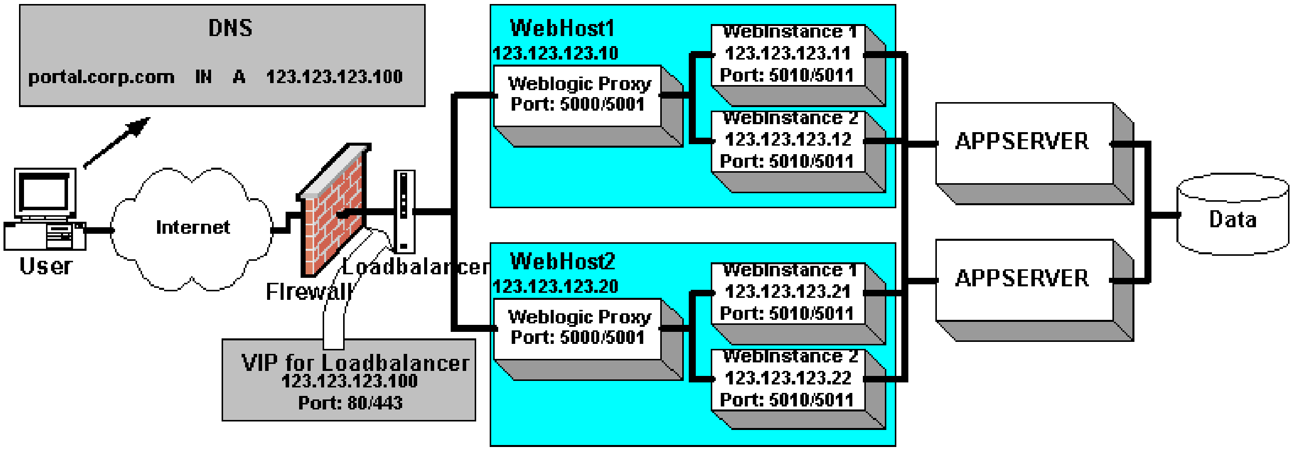
Webserver Setup

Unit	**WebHost1:Instance1** **WebLogic instance name = PIA1**	**WebHost1:Instance2** **WebLogic instance name = PIA2**	**WebHost2:Instance1** **WebLogic instance name = PIA3**	**WebHost2:Instance2** **WebLogic instance name = PIA4**
IP Address1	123.123.123.11[1]	123.123.123.12[1]	123.123.123.21[1]	123.123.123.22[1]
HTTP Port	5010	5010	5010	5010
HTTPS Port	5011	5011	5011	5011

[1] All IP addresses are IP aliases created on Interface 1

Advanced WebLogic Cluster

Use this architecture only if you cannot use the Generic Webserver Cluster discussed in a following section. This architecture improves the Simple WebLogic cluster by removing the single point of failure of the WebLogic proxy server. High Availability is achieved by using a load balancer. The architecture diagram is shown below:



For this setup:

- Use WebLogic proxy serving HttpClusterServlet web application.
- Use Loadbalancer with loadbalancer-cookie based session sticky for HTTP and SSL based session sticky for HTTPS
- All instances must use the same HTTP/HTTPS port e.g. 5010/5011 in the example. This requires IP Aliasing if the host does not have dedicated network interfaces for each webserver instance
- One proxy server load balances across all the WebLogic instances on that host only, so you will use a different multicast address for each host in the cluster
- The proxy will only pass all calls to the webserver instances.
- WebLogic proxy server and WebLogic servers providing PIA are run as WebLogic Managed Servers

Advantages:

- Good Scalability
- Good fault tolerance
- Good flexibility of adding/removing servers dynamically
- Works with non identical webhosts

Disadvantages:

- Proxy adds an additional layer
- WebLogic proxy requires IP Aliasing for running multiple instances
- Reasonably complex to setup

Setup

Refer to the “Big Picture” chapter for other components. To make it easier to understand the architecture illustration shows the mapping for system where the firewall or loadbalancer is not performing any NAT. It is recommended that NAT be performed for greater security.

Proxy Setup

Unit	**WebHost1** **WebLogic instance name = RPS1**	**WebHost2** **WebLogic instance name = RPS2**
IP Address (no NAT[1])	123.123.123.10	123.123.123.20
IP Address (NAT[2])	10.0.0.10	10.0.0.20
Subnet Mask	255.255.255.0	255.255.255.0
Default Route (no Nat[1])	123.123.123.5	123.123.123.5
Default Route (NAT[2])	10.0.0.10	10.0.0.20
HTTP Port	5000	5000
HTTPS Port	5001	5001

[1] Neither Firewall nor Loadbalancer NAT is performed

[2] Either Firewall or Loadbalancer or both (possible but not shown in this document) NAT is performed

Webserver Setup

Unit	**WebHost1:Instance1** **WebLogic**	**WebHost1:Instance2** **WebLogic**	**WebHost2:Instance1** **WebLogic**	**WebHost2:Instance2** **WebLogic**

	instance name = PIA1	**instance name = PIA2**	**instance name = PIA3**	**instance name = PIA4**
IP Address1 (no NAT[2])	123.123.123.11[1]	123.123.123.12[1]	123.123.123.21[1]	123.123.123.22[1]
IP Address1 (NAT[3])	10.0.0.11[1]	10.0.0.12[1]	10.0.0.21[1]	10.0.0.22[1]
HTTP Port	5010	5010	5010	5010
HTTPS Port	5011	5011	5011	5011

[1] All IP addresses are IP aliases created on Interface 1

[2] Neither Firewall nor Loadbalancer NAT is performed

[3] Either Firewall nor Loadbalancer or both (possible but not shown in this document) NAT is performed

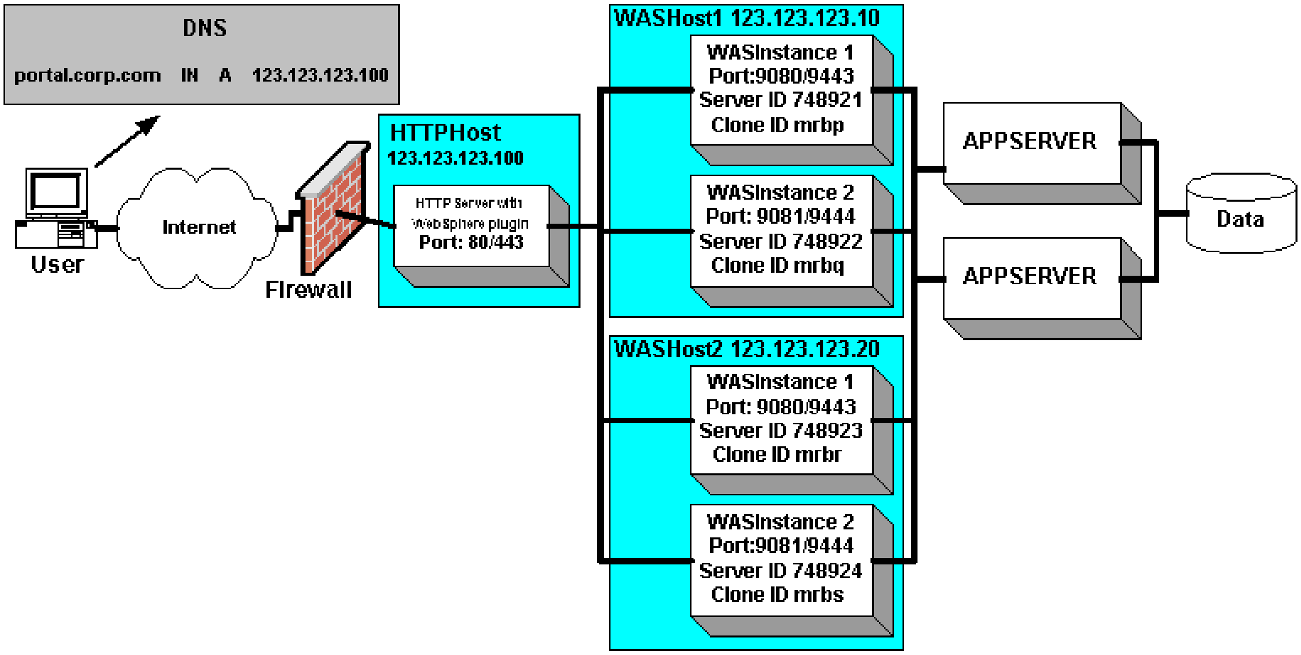
WEBSPHERE CLUSTER

This section only provides specific instructions of how to setup a WebSphere cluster for a PeopleSoft system. Please consult the release notes for WebSphere 4.02 that contains detailed instructions on how to create multiple WebSphere instances on a machine. The instructions in this document are based on those release notes. Download the Release notes and read a file called rncoexist.html. The release notes can be downloaded from ftp://ftp.software.ibm.com/software/websphere/appserv/support/fixpacks/was40/fixpack2/docs/RelNotes402.zip. Note that the architecture proposed here has been adapted for PeopleSoft applications and is therefore somewhat different than the documents at the IBM site.

In the following subsections we discuss two clustering architectures. The first, Simple WebSphere Cluster, has been mentioned here because some systems have been deployed with it. It is the simplest to setup but is also the least scalable and is only applicable to small sites. The second, Advanced WebSphere Cluster, is more scalable and is recommended for customers with high scalability requirement. However, for the most scalable and flexible architecture the Generic Webserver Cluster described in the next section is recommended.

Simple WebSphere Cluster

In a simple WebSphere cluster there are one or more WebSphere instances (WASInstance) per webserver host (WASHost) and there are more than one webserver host for redundancy. There is only one HTTP proxy server running on a separate host. The architecture diagram is shown below:



For this setup:

- Use an HTTP server with the WebSphere plug-in installed
- The WebSphere instances may listen to ports other than 9080 and 9081 in the example.
- Requests are workload managed across the WebSphere instances by the WebSphere plug-in on HTTPHost. The WebSphere plug-in runs within the HTTP server process.
- All requests are passed to the WebSphere instances. No content is served from the HTTPHost.

Advantages:

- Simple to setup
- Low cost solution
- WebSphere instances can listen on any port, which does not require IP aliasing.

Disadvantages:

- HTTP Server (proxy) adds an additional layer
- HTTP Server (proxy) is a single point of failure.

- Very low scalability. The HTTP server will soon become a scalability bottleneck as new webservers are added to the pool particularly when going over SSL.

Setup

The setup is considerably different from the “Big Picture” chapter because the load balancer is missing. To make it easier to understand the architecture illustration shows the mapping for a system where the firewall is not performing any NAT. It is recommended that NAT be performed for greater security.

HTTPHost (proxy) Setup

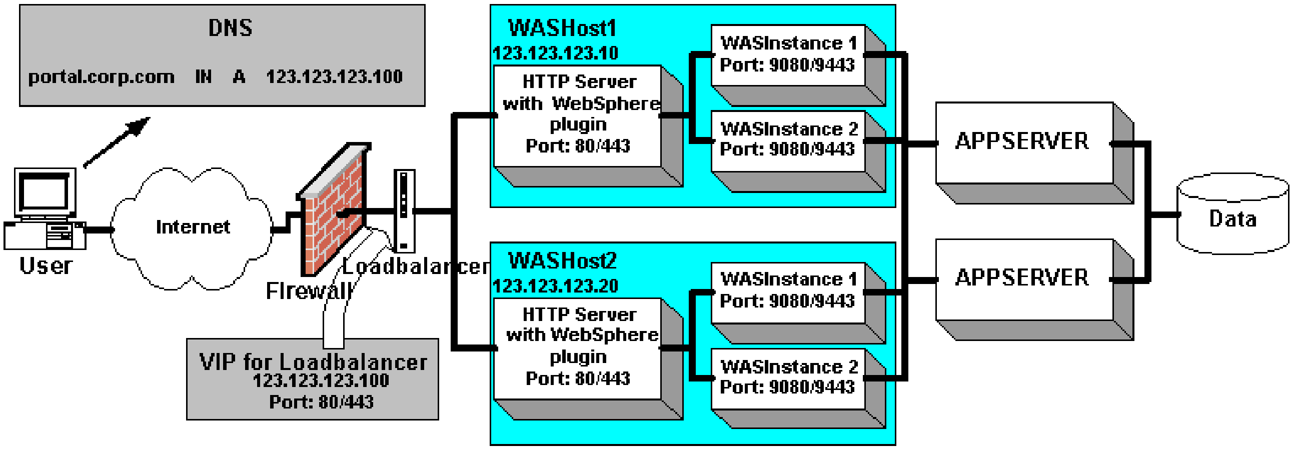
Unit	HTTPHost
IP Address	123.123.123.100
Subnet Mask	255.255.255.0
Default Route	123.123.123.1
HTTP Port	80
HTTPS Port	443

Webserver Setup

Unit	WASHost1:Instance1	WASHost1:Instance2	WASHost2:Instance1	WASHost2:Instance2
IP Address1	123.123.123.11	123.123.123.12	123.123.123.21	123.123.123.22
HTTP Port	9080	9081	9080	9081
HTTPS Port	9443	9444	9443	9444

Advanced WebSphere Cluster

Use this architecture only if you cannot use the Generic Webserver Cluster discussed in the next section. This architecture improves the Simple WebSphere cluster by removing the single point of failure of the HTTPHost (proxy) server. High Availability is achieved by using a load balancer. The architecture diagram is shown below:



For this setup:

- Use HTTP Server (proxy)
- Use Loadbalancer with loadbalancer-cookie based session sticky for HTTP and SSL based session sticky for HTTPS
- One HTTP Server (proxy) server load balances across all the WebSphere instances on that host

Advantages:

- Good Scalability
- Good fault tolerance. WebSphere plug-in provides failover support. The WebSphere plug-in will forward requests to available WASInstances. If a WASInstance is down, the plug-in will temporarily route around it until the instance becomes available again.
- Good flexibility of adding/removing servers dynamically
- Works with non identical webhosts

Disadvantages:

- HTTP Server (proxy) adds an additional layer
- Reasonably complex to setup with added complexity of loadbalancer setup

Setup

Refer to the “Big Picture” chapter for other components. To make it easier to understand the architecture illustration shows the mapping for system where the firewall or loadbalancer is not performing any NAT. It is recommended that NAT be performed for greater security.

HTTP Server (proxy) Setup

Unit	HTTP server on WASHost1	HTTP server on WASHost2
IP Address (no NAT[1])	123.123.123.10	123.123.123.20
IP Address (NAT[2])	10.0.0.10	10.0.0.20
Subnet Mask	255.255.255.0	255.255.255.0
Default Route (no Nat[1])	123.123.123.5	123.123.123.5
Default Route (NAT[2])	10.0.0.10	10.0.0.20
HTTP Port	80	80
HTTPS Port	443	443

[1] Neither Firewall nor Loadbalancer NAT is performed

[2] Either Firewall or Loadbalancer or both (possible but not shown in this document) NAT is performed

Webserver Setup

Unit	WASHost1:Instance1	WASHost1:Instance2	WASHost2:Instance1	WASHost2:Instance2
IP Address1 (no NAT[1])	123.123.123.10	123.123.123.10	123.123.123.20	123.123.123.20
IP Address1 (NAT[2])	10.0.0.10	10.0.0.10	10.0.0.20	10.0.0.20
HTTP Port	9080	9081	9080	9081
HTTPS Port	9443	9444	9443	9444

[1] Neither Firewall nor Loadbalancer NAT is performed

[2] Either Firewall or Loadbalancer or both (possible but not shown in this document) NAT is performed

ORACLE APPLICATION SERVER (OAS) CLUSTER

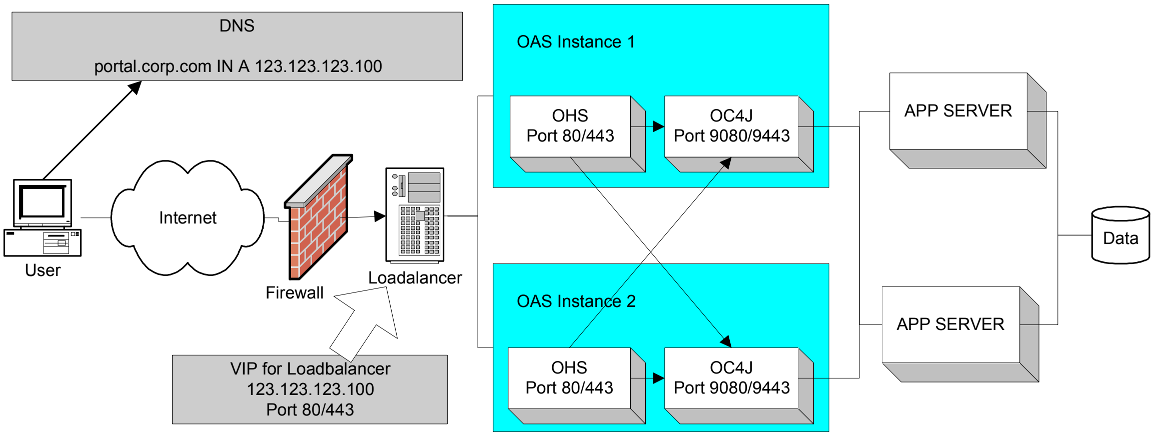
This section provides specific instructions of how to setup an OAS 10.1.2.x cluster for a PeopleSoft 8.47 – 8.48 releases and OAS 10.1.3.1 for 8.49 releases.

Note:
Please refer to OAS documentation (http://download-west.oracle.com/docs/cd/B14099_11/index.htm) for a detailed understanding of OAS 10.1.2.x clustering and (http://download-west.oracle.com/docs/cd/B31017_01/index.htm) for a detailed understanding of OAS 10.1.3.1 clustering

OAS does not support Simple Cluster since the use of OHS component with OAS is mandatory.

Advanced OAS Cluster

High Availability is achieved by using a load balancer. The architecture diagram is shown below:



For this setup:

- Use HTTP Server (proxy)
- Use Loadbalancer with loadbalancer-cookie based session sticky for HTTP and SSL based session sticky for HTTPS
- One HTTP Server (proxy) server load balances across all the OAS instances on that host

Advantages:

- Good Scalability

- Good fault tolerance.
- Good flexibility of adding/removing servers dynamically
- Works with non identical webhosts

Disadvantages:

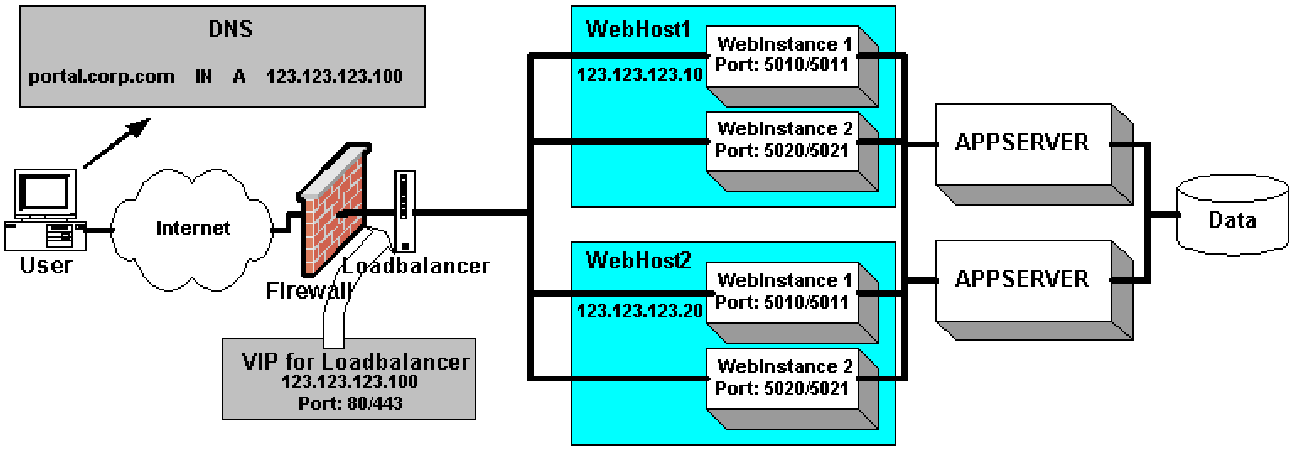
- HTTP Server (proxy) adds an additional layer
- Reasonable complex to setup with added complexity of loadbalancer setup

GENERIC WEBSERVER CLUSTER

Unlike WebLogic/WebSphere clustering architecture there is only one form of Generic Webserver Cluster architecture. The capacity of the system will determine the selection of components but the over all architecture does not change.

Generic Webserver Cluster

In a generic webserver cluster there is no need to install WebLogic proxy server or a WebShpere HTTP server and therefore one layer of indirection can be avoided. There is one or more WebLogic/WebSphere instance per webserver host and there is more than one webserver host for redundancy. This configuration is recommended for sites which require the ability to change capacity on demand. All web hosts need not be identical and load can be distributed according to host capacity (with most loadbalancers). High scalability is achieved by using HW loadbalancer to distribute the load directly across all the webserver instances. This is by far the most flexible and scalable configuration. The architecture diagram is shown below:



For this setup:

  - Use HW Loadbalancer with loadbalancer-cookie based session sticky or HTTP and SSL based session sticky for HTTPS.
  - The webserver instances need not run on the same port numbers e.g. 5010/5011, 5020/5021 in the example.

Advantages:

- Easy to setup
- Webserver instances can be installed on any port. This enables us to run multiple instances of the webserver on a host without using IP Aliasing
- Best Scalability
- Best fault tolerance
- Best flexibility of adding/removing servers dynamically

Disadvantages:

- For clusters using WebSphere webservers - internal HTTP server does not have logging for requests. Traffic analysis may not be possible without standard request logs

Setup

Refer to the “Big Picture” chapter for other components. To make it easier to understand the architecture illustration shows the mapping for system where the firewall or loadbalancer is not performing any NAT. It is recommended that NAT be performed for greater security.

Webserver Setup

Unit	WebHost1:Instance1	WebHost1:Instance2	WebHost2:Instance1	WebHost2:Instance2
IP Address (no NAT[1])	123.123.123.10	123.123.123.10	123.123.123.20	123.123.123.20
IP Address (NAT[2])	10.0.0.10	10.0.0.10	10.0.0.20	10.0.0.20
HTTP Port	5010 or 9080	5020 or 9081	5010 or 9080	5020 or 9081
HTTPS Port	5011 or 9443	5021 or 9444	5011 or 9443	5021 or 9444

[1] Neither Firewall nor Loadbalancer NAT is performed

[2] Either Firewall or Loadbalancer or both (possible but not shown in this document) NAT is performed

Configuring a WebLogic Proxy Server

This section applies to the Simple WebLogic Clustering and Advanced WebLogic Clustering architecture only. This section describes the steps to configure a ***WebLogic*** proxy server for clustering.

IIS Proxy Plug-in

If you are using the IIS proxy plug-in please follow the PeopleSoft installation guide to configure the plug-in as a proxy server. For PeopleTools 8.44 and higher please refer to the “Working with WebLogic” section of the “PeopleTools: Server Tools” PeopleBook.   Once installation is complete and has been tested to function with one instance of the ***WebLogic*** servers follow the following steps to cluster enable the proxy server by editing the iisproxy.ini file.

Here is a sample iisproxy.ini file without clustering. Comment lines are denoted with the "#" (hatch) character.

```
# This file contains initialization name/value pairs
# for the IIS/WebLogic plug-in.
WebLogicHost=ProxyHost1
WebLogicPort=5010
ConnectTimeoutSecs=20
ConnectRetrySecs=2
WLCookieName=PORTAL-PSJSESSIONID
```

Here is a sample iisproxy.ini file with clustering. Comment lines are denoted with the "#" (hatch) character.

```
# This file contains initialization name/value pairs
# for the IIS/WebLogic plug-in.
WebLogicCluster= WebHost1:5010,WebHost2:5010
ConnectTimeoutSecs=20
ConnectRetrySecs=2
WLCookieName=PORTAL-PSJSESSIONID
```

where WLCookieName must match the property `portalServletSessionCookieName` and `CookieName` in `PS_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL\WEB-INF\weblogic.xml` and in your Web Profile in your database for PeopleTools 8.49 and below.

For PeopleTools 8.50 and above, WLCookieName must match the property `CookieName in PIA_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL.war\WEB-INF\weblogic.xml`

Optional: Create a DNS name entry for the IP address for the proxy i.e. `123.123.123.10,` that will be used to access the website.

SunOne (Also known as Sun Java System WebServer , and formally known as iPlanet , Netscape) Proxy Plug-in

If you are using the SunOne proxy plug-in please follow the PIA Answerbook to configure the plug-in as a proxy server. For PeopleTools 8.44 and above please refer to the “Working with WebLogic” section of the “PeopleTools: Server Tools” in PeopleBook. Once installation is complete and has been tested to function with one instance of the WebLogic servers make the highlighted change to cluster enable the proxy server by editing the obj.conf file. The file has been taken from the sample posted in PIA Answerbook and updated for clustering.

The following example illustrates sample parameters in the obj.conf file.

```
# Use only forward slashes in pathnames--backslashes can
cause problems. See the iPlanet
# documentation for more information.
# This obj.conf will proxy requests for based on URLS
that contain either ‘peoplesoft8’ or
# ‘servlets’ or both.

Init fn="load-modules" funcs="wl-proxy,wl-init"\
 shlib=C:/Netscape/Server4/plugins/proxy36.dll
Init fn="wl-init"
Init fn=flex-init access="C:/Netscape/Server4/https-
DBROWN032500.peoplesoft.com/logs/access"
format.access="%Ses->client.ip% - %Req->vars.auth-user%
[%SYSDATE%] \"%Req->reqpb.clf-request%\" %Req-
>srvhdrs.clf-status% %Req->srvhdrs.content-length%"
Init fn=load-types mime-types=mime.types
Init fn="load-modules"
shlib="C:/Netscape/Server4/bin/https/bin/NSServletPlugin.
dll"
funcs="NSServletEarlyInit,NSServletLateInit,NSServletName
Trans,NSServletService" shlib_flags="(global|now)"
Init fn="NSServletEarlyInit" EarlyInit=yes
Init fn="NSServletLateInit"  LateInit=yes

<Object name=default>
NameTrans fn="NSServletNameTrans" name="servlet"
NameTrans fn="pfx2dir" from="/servlet"
dir="C:/Netscape/Server4/docs/servlet"
name="ServletByExt"
```

```
NameTrans fn=pfx2dir from=/ns-icons
dir="C:/Netscape/Server4/ns-icons" name="es-internal"
NameTrans fn=pfx2dir from=/mc-icons
dir="C:/Netscape/Server4/ns-icons" name="es-internal"
NameTrans fn="pfx2dir" from="/help"
dir="C:/Netscape/Server4/manual/https/ug" name="es-
internal"
NameTrans fn="pfx2dir" from="/manual"
dir="C:/Netscape/Server4/manual/https" name="es-internal"
NameTrans fn=document-root
root="C:/Netscape/Server4/docs"
PathCheck fn=nt-uri-clean
PathCheck fn="check-acl" acl="default"
PathCheck fn=find-pathinfo
PathCheck fn=find-index index-
names="index.html,home.html"
ObjectType fn=type-by-extension
ObjectType fn=force-type type=text/plain
Service method=(GET|HEAD) type=magnus-internal/imagemap
fn=imagemap
Service method=(GET|HEAD) type=magnus-internal/directory
fn=index-common
Service method=(GET|HEAD) type=*~magnus-internal/*
fn=send-file
AddLog fn=flex-log name="access"
</Object>

<Object name="peoplesoft8" ppath="*/peoplesoft8/*">
Service fn=wl-proxy \
 WebLogicCluster="WebHost1:5010,WebHost2:5010"
 WLCookieName=PORTAL-PSJSESSIONID
</Object>

<Object name="servlets" ppath="*/servlets/*">
Service fn=wl-proxy \
      WebLogicCluster="WebHost1:5010,WebHost2:5010"
      WLCookieName=PORTAL-PSJSESSIONID
</Object>

<Object name=cgi>
ObjectType fn=force-type type=magnus-internal/cgi
Service fn=send-cgi
</Object>

<Object name="servlet">
ObjectType fn=force-type type=text/html
Service fn="NSServletService"
</Object>

<Object name="jsp092">
ObjectType fn="type-by-extension"
ObjectType fn="change-type" type="magnus-internal/jsp092"
if-type="magnus-internal/jsp"
```

```
Service fn="NSServletService" type="magnus-
internal/jsp092"
</Object>

<Object name="ServletByExt">
ObjectType fn=force-type type=magnus-internal/servlet
Service type="magnus-internal/servlet"
fn="NSServletService"
</Object>

<Object name="es-internal">
PathCheck fn="check-acl" acl="es-internal"
</Object>
```

where WLCookieName must match the property `portalServletSessionCookieName` and `CookieName` in `WL_HOME\config\peoplesoft\applications\PORTAL\WEB-INF\psftdocs\ps\configuration.properties` file and `WL_HOME\config\peoplesoft\applications\PORTAL\WEB-INF\weblogic.xml` file respectively. For PT8.44 –PT8.49, the path is different and it is `PS_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL\WEB-INF\psftdocs\ps\configuration.properties` and `PS_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL\WEB-INF\weblogic.xml` respectively.

For PT8.50 and higher, the path is `PIA_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL.war\WEB-INF\psftdocs\ps\configuration.properties` and `PIA_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL.war\WEB-INF\weblogic.xml` respectively.

Optional: Create a DNS name entry for the IP address for the proxy i.e. `123.123.123.10`, that will be used to access the website.

Apache Proxy Plug-in

If you are using the Apache HTTP proxy plug-in please follow the PIA Answerbook to configure the plug-in as a proxy server. For PeopleTools 8.44 and higher please refer to the “Working with WebLogic” section of the “PeopleTools: Server Tools” PeopleBook. Once installation is complete and has been tested to function with one instance of the WebLogic servers make the highlighted change to cluster enable the proxy server by editing the httpd.conf file. The file has been taken from the sample posted in PIA Answerbook and updated for clustering.

Specify the parameters that will be used by the Apache plugin by defining them in an IfModule tag for WebLogic in your Apache's httpd.conf. This tag should be added in the "### Section 2: 'Main' server configuration" section of your httpd.conf.

```
<IfModule mod_weblogic.c>
     WebLogicCluster WebHost1:5010,WebHost2:5010
     WLCookieName=PORTAL-PSJSESSIONID
</IfModule>
```

where WLCookieName must match the property portalServletSessionCookieName and CookieName in WL_HOME\config\peoplesoft\applications\PORTAL\WEB-INF\psftdocs\ps\configuration.properties and WL_HOME\config\peoplesoft\applications\PORTAL\WEB-INF\weblogic.xml file respectively. . For PT8.44 – PT8.49, the path is different and it is PS_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL\WEB-INF\psftdocs\ps\configuration.properties and PS_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL\WEB-INF\weblogic.xml respectively.

For PT8.50 and higher, the path is PS_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL.war\WEB-INF\psftdocs\ps\configuration.properties and PS_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL.war\WEB-INF\weblogic.xml respectively.

Optional: Create a DNS name entry for the IP address for the proxy i.e. 123.123.123.10, that will be used to access the website.

WebLogic server as Proxy

Please note: Using WebLogic Server as Proxy (HttpClusterServlet) in the Production Environment is not recommended per BEA/Oracle. It is only recommended to be used in staging environments or development environments.

PeopleTools 8.40 – 8.43

The instructions below assumes that WebLogic is up and running on the host and on NT/2000 or windows 2003 it can be started from the command line, please consult PIA Answerbook for detailed instructions on how to do this. It is necessary to install PIA on this host even if no instance of PIA will run on this host. In the instructions below, WL_HOME is the directory where WebLogic is installed and PIA1 is the name of the first WebLogic name. The steps to follow are:

1. Backup the WL_HOME/config/peoplesoft/config.xml file
2. In a browser navigate to the console at http://webhost1:5000/console

NOTE: The default login is - system/password

3. Select `peoplesoft->Deployments` from the left frame
4. Select “`Web Applications`” from the left frame
5. Select “`Configure a new Web Application...`” from the right frame
6. Specify the following information:
    a. From `Configuration->General` Tab
        - Change Name to “`HttpClusterServlet`”
        - Change Path URI to `config/peoplesoft/applications/HttpClusterServlet`
        - Select Create
7. Select `peoplesoft->Servers` from the left frame
8. Select “`Configure a new Server...`” from the right frame
9. Modify data to match the Proxy Server values. In our example:
    a. From Configuration->General Tab
        - Change Name to `RPS`
        - Change Listen Port to `5000`
        - Specify Listen Address if the host has multiple IP addresses
        - Select Apply
    b. From `Configuration->HTTP` Tab
        - Select Default Web Application as “`HttpClusterServlet`”
        - Select Apply
    c. Optional from `Configuration->SSL` Tab
        - Change Port to `5001`
        - Setup “`Server Key File Name`”
        - Setup “`Server Certificate File Name`”
        - Setup “`Server Certificate Chain File Name`”
        - Setup “`Trusted CA File Name`”
        - Select Apply
    d. From `Deployments->Web Applications` Tab
        - Select “`Web App Component`” and then select “`HttpClusterServlet`” as a Chosen item
        - Select Apply
    e. From `Logging->General` Tab
        - Change “`File Name`” to create a new log file for RPS instance, e.g.

`WL_HOME/config/peoplesoft/logs/RPS_weblogic.log`
  - Select Apply

f. From `Logging->HTTP` Tab
  - Change "`Logfile Name`" to create a new access log file for RPS instance e.g. `WL_HOME/config/peoplesoft/logs/RPS_access.log`
  - Select Apply

g. Optionally from the `Notes` Tab
  - Change Notes to something like:
"`The RPS server is the is the Reverse Proxy Server to access PIA instances`"

10. Select `peoplesoft->Deployments` from the left frame
11. Then Select `Web Applications` from the left frame
12. Select "`Edit Web Application Descriptor…`" from the right frame
13. Select "`HttpClusterServlet->Web App Descriptor->Servlets->HttpClusterServlet->Parameters`" from the left frame of the new window
14. Then for WebLogic 6.1 SP2 select "`WebLogicCluster`" or for WebLogic 6.1 SP1 select "`defaultServers`" from the left frame
    a. Update the "`Param Value`" field to include all webserver instance info (name:httpPort:httpsPort) here.
    `WebHost1:5010:5011|WebHost2:5010:5011`
    b. Select Apply
15. Select "HttpClusterServlet->Web App Descriptor->Servlets->HttpClusterServlet->Parameters" from the left frame
16. Click on "`Configure a new parameter...`" on the right frame
17. Specify the following information:
    a. Set `Param-Name: cookieName`
    b. Set `Param-Value: PORTAL-PSJSESSIONID`
    NOTE: With WebLogic 6.1 SP1 the only valid Param-Value is `JSESSIONID` which is also the default.
    NOTE: The Param-Value must match the `CookieName` in PORTAL's `weblogic.xml` and `configuration.properties` files. On NT these are `WL_HOME\config\peoplesoft\applications\PORTAL\WEB-INF\weblogic.xml` and `WL_HOME\config\peoplesoft\applications\PORTAL\WEB-INF\psftdocs\ps\configuration.properties` respectively.
18. Select `HttpClusterServlet` on the left frame
    a. Select Validate on the right frame and ensure that the deployment descriptor validates cleanly
19. Select HttpClusterServlet on the left frame again
    a. Select Persist to save the validated deployment descriptor
20. On NT create a startup file `WL_HOME\config\peoplesoft\startRPS.cmd` with the following:

```
@echo off
startPIA RPS
```

On UNIX create a startup file `WL_HOME\config\peoplesoft\startRPS.sh` with the following:

```
#!/bin/sh
exec startPIA.sh RPS
```

21. **Optional:** Create a DNS name entry for the IP address for the proxy i.e. `123.123.123.10,` that will be used to access the website.

This will configure the WebLogic proxy. After these steps are completed use `WL_HOME\config\peoplesoft\startRPS.cmd (.sh)` to start the WebLogic proxy.

PeopleTools 8.44 – 8.48

The Multi-server option of the PIA install builds a WebLogic proxy server for you. The WebLogic instance name is RPS and the two applications deployable to it are HttpProxyServlet and HttpClusterServlet. HttpProxyServlet is used to proxy content from a single back-end WebLogic instance. HttpClusterServlet is used to proxy content from multiple back-end WebLogic servers. Refer to the following section titled “Configuring Multiple WebLogic instances” for information on the usage of the HttpClusterServlet.

The back-end WebLogic content servers that the WebLogic proxy will proxy content from is defined in the web.xml of HttpProxyServlet/HttpClusterServlet web applications. To edit these settings use WebLogic Builder.

Note: startWebLogicBuilder is a graphical java application which mean on UNIX, X11 libraries must be installed and an xterm must be available.

To start WebLogic Builder run either of the following commands from within your WebLogic domain directory. Once WebLogic Builder starts click File/Open to specify which application’s descriptors to edit or specify the application name on the command line as shown below with HttpClusterServlet. The application name is case sensitive.

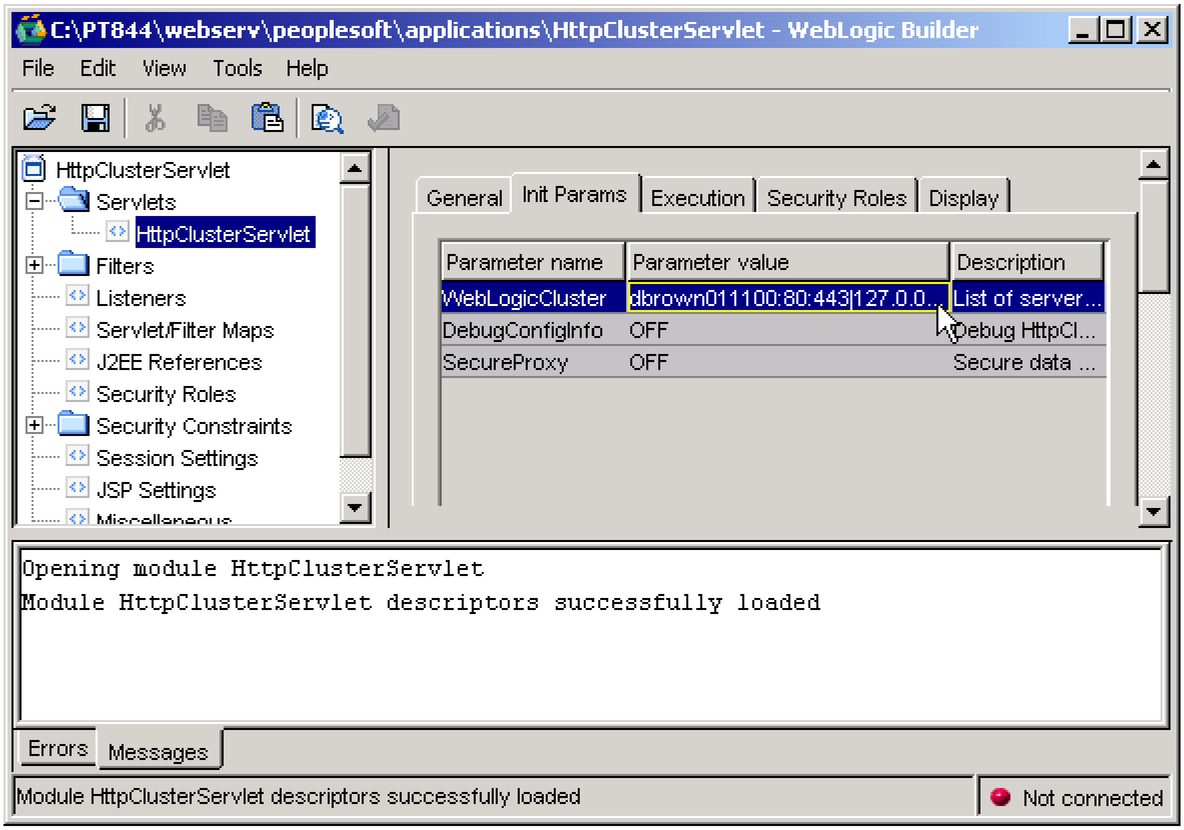
1. startWebLogicBuilder applications/HttpClusterServlet

or

1. setEnv (on UNIX run . ./setEnv.sh)
2. java weblogic.marathon.Main applications/HttpClusterServlet -stddialog

To edit the list of content server(s) the proxy server will proxy content fromselect ‘Servlets/”*proxyServlet” in the Navigation window on the left. Next click on the “Init Params” tab and select either WebLogicCluster, WebLogicHost or WebLogicPort. If you are editing the HttpClusterServlet you will see the WebLogicCluster parameter, otherwise if you are editing the HttpProxyServlet you will see WebLogicHost and WebLogicPort.

For example;



Parameter name	Parameter value	Description
WebLogicCluster	dbrown011100:80:443|127.0.0...	List of server...
DebugConfigInfo	OFF	Debug HttpCl...
SecureProxy	OFF	Secure data ...

```
Opening module HttpClusterServlet
Module HttpClusterServlet descriptors successfully loaded
```

Module HttpClusterServlet descriptors successfully loaded

After your changes are complete, click File/Save and restart your WebLogic Proxy server if it was running for the changes to take effect.

With the PIA install, the RPS server deploys “HttpProxyServlet” as default application and we want to deploy “HttpClusterServlet” when cluster is configured. In order to make this change, login to WebLogic admin console and follow the steps listed below.

1. Select `peoplesoft->Deployments` from the left frame
2. Then select HttpProxyServlet from the list of current deployments in the right frame
3. Select Target tab and uncheck the “RPS” server from the list of independent servers table in the right frame. Click on “Apply” button.
4. Now go back to deployments tab in the left frame and HttpClusterServlet from the list of current deployments in the right frame.
5. Select Target tab and check the “RPS” server from the list of servers. Click on “Apply” button.

Once these changes are done in the console, restart the RPS server in order to have the changes to take effect.

PeopleTools 8.49 – PeopleTools 8.50

WebLogic Server version 9.2 is supported on PT 8.49 and WebLogic Server version 10.3.1 is supported on PT8.50. Starting WebLogic 9.2, the application startWebLogicbuilder is deprecated. To configure WebLogic, use WebLogic console along with the configuration files. In order to configure WebLogic as a reverse proxy server, follow the procedure listed below.

Please note: For WebLogic Server Version 10.3.1 (For PT8.50) It is required to apply an additional One-Off-Patch -Bug8348275_1031GA.jar- from BEA in order to support WLCookieName in HttpClusterServlet. Please refer to "PeopleSoft Enterprise PeopleTools PT 8.50 WebLogic Patches" for detail on how to obtain and apply this One-Off-Patch.

Setup HttpClusterServlet as the target for RPS Server:

After the multiserver PIA installation is done, login to WebLogic console using http://weblogichost:9999/console and follow the steps listed below.

- Click on PeopleSoft, Deployments and select HttpClusterServlet web application
- Click on Targets tab and make sure the server "RPS" is selected as Target and click Save button
- Click on PeopleSoft, Deployment and select HttpProxyServlet web application
- Click on Targets tab and make sure all the servers are **unchecked** for this application

Setup list of WebLogic servers in the cluster:

The back-end WebLogic content servers that the WebLogic proxy will proxy content from is defined in the web.xml of HttpProxyServlet/HttpClusterServlet web applications.

To change the settings in PT849, edit web.xml available in PS_HOME\webserv\<domain>\applications\HttpClusterServlet\WEB-INF.

To change the settings in PT850, edit web.xml available in PIA_HOME\webserv\<domain>\applications\HttpClusterServlet\WEB-INF.

- Backup the existing web.xml
- Edit web.xml, find the following section and enter the list of server names for "WebLogicCluster" parameter as it describes in the description tag.

```
<servlet>

   <display-name>HttpClusterServlet</display-name>

   <servlet-name>HttpClusterServlet</servlet-name>

   <servlet-class>weblogic.servlet.proxy.HttpClusterServlet</servlet-class>

   <init-param>

            <description>List of servers in the cluster.  Seperate by '|'</description>
```

```
    <param-name>WebLogicCluster</param-name>

    <param-value>host1name:6666:6667|host2name:6666:6667</param-value>

  </init-param>
```

- Add WLCookieName parameter to the same web.xml.

```
<init-param>

    <param-name>WLCookieName</param-name>

     <param-value>PORTAL-PSJSESSIONID</param-value>

</init-param>
```

where **WLCookieName** must match the property portalServletSessionCookieName and CookieName in PS_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL\WEB-INF\psftdocs\ps\configuration.properties and PS_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL\WEB-INF\weblogic.xml respectively for PT8.49.

For PT8.50, the path is PIA_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL.war\WEB-INF\psftdocs\ps\configuration.properties and PIA_HOME\webserv\<weblogic_domain>\applications\peoplesoft\PORTAL.war\WEB-INF\weblogic.xml respectively.

- Once the parameter is modified, save the file and restart the RPS server

CONFIGURING MULTIPLE WEBLOGIC INSTANCES

This section applies to all clustering architecture that uses WebLogic. In addition, if you are configuring SimpleWebLogic Clustering or Advanced WebLogic Clustering, you will also need to configure a proxy server as described in the previous section. This section describes the steps to configure multiple instances of webservers. First, follow PeopleSoft Internet Architecture (PIA) installation instructions to configure and test WebLogic on the webserver host. The instructions below assumes that WebLogic is up and running on the host and on MicroSoft Windows OS, and it can be started from the command line.

PIA1 is the first webserver instance name. The steps to follow are:

PeopleTools 8.40 – 8.43

In the instructions below `WL_HOME` is the directory where WebLogic is installed.

1. Backup the `WL_HOME/config/peoplesoft/config.xml` file
2. In a browser navigate to the console at http://webhost1:5000/console

   NOTE: The default login is - system/password
3. Select `peoplesoft->Servers` from the left frame
4. Select "`PIA`" from the left frame
5. Modify data to match the instance1 values. In our example :
   a. From Configuration->General Tab
      - Change Listen Port to `5010`
      - Change Listen Address to the IP address this WebLogic instance should bind to
      - Select Apply

   b. Optional from `Configuration->SSL` Tab
      - Change Port to `5011`
      - Select Apply

   c. From `Logging->General` Tab
      - Change "`File Name`" to create a new log file for instance1 to `WL_HOME/config/peoplesoft/logs/PIA1_weblogic.log`
      - Select Apply

   d. From `Logging->HTTP` Tab
      - Uncheck "`WAP Enabled`" required to workaround WebLogic bug
      - Change "`Logfile Name`" to create a new access log instance1 `WL_HOME/config/peoplesoft/logs/PIA1_access.log`
      - Select Apply

   e. Optionally from the `Notes` Tab
      - Change Notes to something like:

      `"The PIA1 server is the original instance of the default PIA server for PeopleSoft Internet Architecture"`
6. On NT create a startup file `WL_HOME\config\peoplesoft\startPIA1.cmd` with the following:

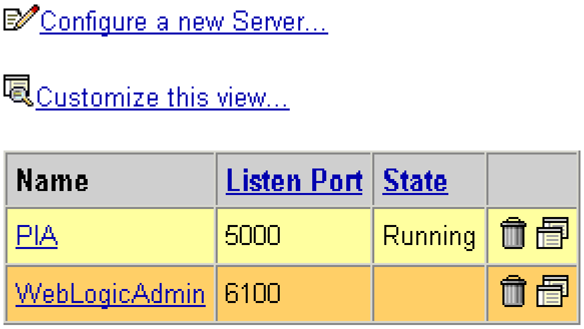
```
    @echo off
    startPIA PIA1
```

On UNIX create a startup file `WL_HOME\config\peoplesoft\startPIA1.sh` with the following:

```
#!/bin/sh
exec startPIA.sh PIA1
```

7. Edit `WL_HOME\config\peoplesoft\config.xml`
   a. Change all reference of “PIA” to “PIA1”
8. This prepares instance1. Continue with the following steps to create additional instances
9. Restart the webserver with `startPIA1.cmd (.sh)` and in a browser navigate to the console at http://webhost1:5000/console

   NOTE: The default login is - system/password
10. Select `peoplesoft->Servers` from the left frame
11. Select the clone option from the right frame for the PIA row. (Clone is the icon with stacked windows next to the trash can)

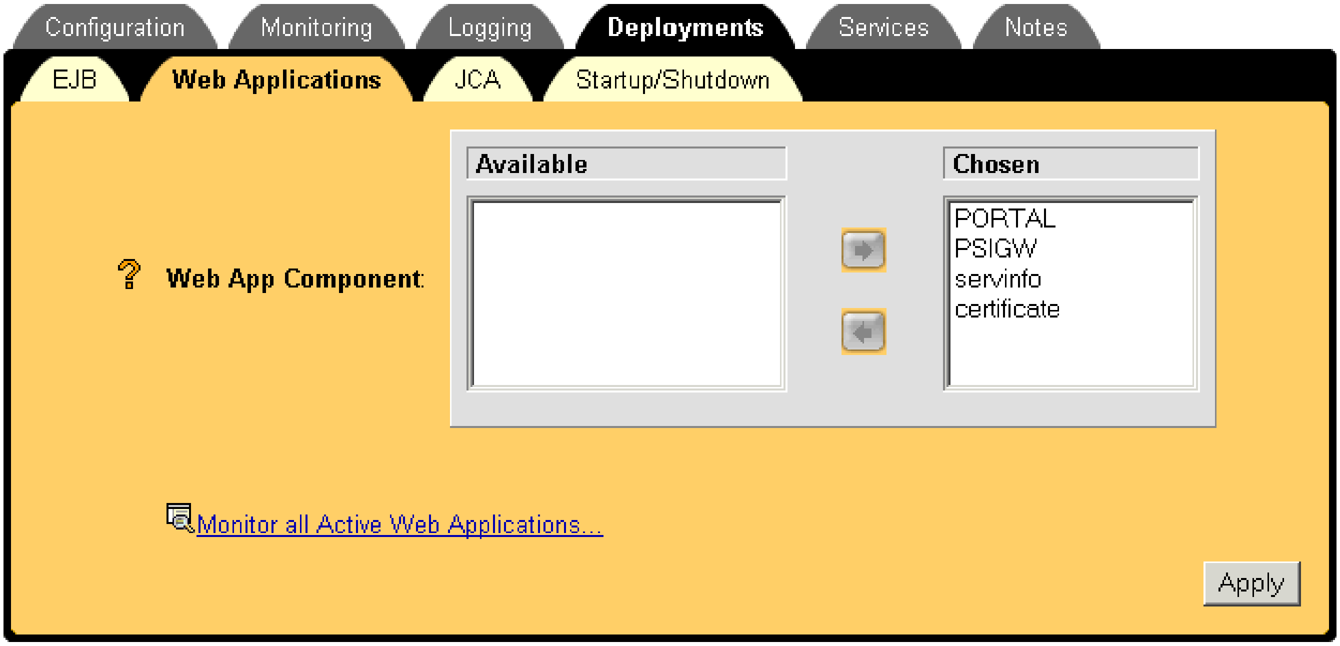


Configure a new Server...

Customize this view...

Name	Listen Port	State	
PIA	5000	Running	🗑 ⧉
WebLogicAdmin	6100		🗑 ⧉

12. Modify the cloned data to match the instance2 values. In our example :
    a. From `Configuration->General` Tab
       - Change `Name` to `PIA2`
       - Change `Listen Port` to `5010`
       - Select Apply
    b. From `Configuration->SSL` Tab
       - Change `Port` to `5011`
       - Select Apply
    c. From `Deployments->Web Applications` Tab
       - Select “`Web App Component`” as shown below



Web App Component:

Available	Chosen
	PORTAL PSIGW servinfo certificate

- Select Apply

d. From `Logging->General` Tab

- Change “`Logfile Name`” to create a new access log file for the cloned instance `WL_HOME/config/peoplesoft/logs/PIA2_weblogic.log`
- Select Apply

e. From `Logging->HTTP` Tab

- Change “`File Name`” to create a new log file for the cloned instance `WL_HOME/config/peoplesoft/logs/PIA2_access.log`
- Select Apply

f. Optionally for the Notes Tab

- Change “`Notes`” to something like: `“The PIA2 server is the cloned instance of the default PIA server for PeopleSoft Internet Architecture.”`

13. On NT create a startup file `WL_HOME\config\peoplesoft\startPIA2.cmd` with the following:

```
@echo off
startPIA PIA2
```

On UNIX create a startup file `WL_HOME\config\peoplesoft\startPIA2.sh` with the following:

```
#!/bin/sh
exec startPIA.sh PIA2
```

NOTE: Webserver instances cannot be brought up all at once. They must be started in sequence. Currently, this has to be implemented by either bringing up the servers manually or by putting sufficient sleep time in the startup files so that only one instance comes up at a time.

14. Edit `WL_HOME\config\peoplesoft\config.xml`
    a. Stop the webserver first.
    b. Change all reference like `"Server-1024176849200"` to `"PIA2"`
    c. Restart the webserver

15. This will configure "PIA2" instance of the webserver. To install additional instances of the webserver, repeat steps 9 through 14, remember to replace "PIA2" with the instance name of the instance being created. Also remember to update the WebLogic proxy's configuration to add these instances along with their http/https ports. For Generic Webserver Clustering step 16 through 19 may be skipped.

16. Next setup a WebLogic cluster. Select `peoplesoft->Clusters` from the left frame

17. Select `"Configure a new Cluster..."` from the right frame

18. Specify the following information:

    a. From `Configuration->General` Tab

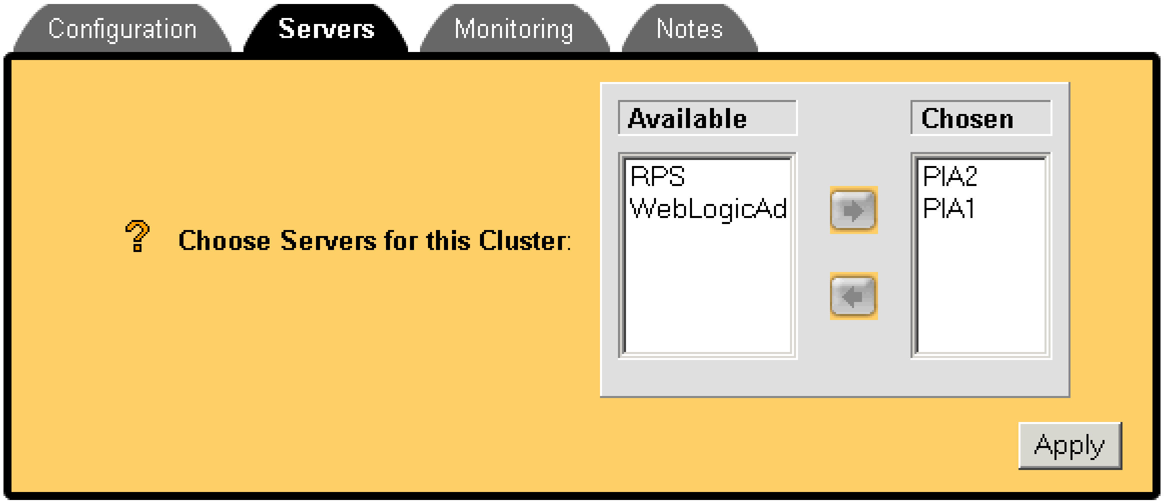
       - Optional - Change Name
       - Set `"Cluster Address"` to the VIP of the virtual name of the cluster, i.e. portal.corp.com
       - Select Create

    b. From `Configuration->Multicast` Tab

       - Ensure the default multicast address is not used on your network. If required change the value and select Apply.

    c. From `Configuration->Servers` Tab

       - Select PIA1, PIA2 and all other instances of PIA that need to be part of this cluster.



Choose Servers for this Cluster:

Available: RPS, WebLogicAd

Chosen: PIA2, PIA1

- Select Apply

d. Optionally, from `Configuration->Notes` Tab

- Add notes
- Select Apply

19. Restart all webservers and proxy server if applicable.

20. **Optional:** If a separate DNS name is configured for the proxy/loadbalancer or an SSL accelerator is used on those devices, it will be necessary to setup **`pswebservername`**, **`defaultScheme`** and **`defaultPort`** in those cases.

21. **Optional:** To configure the webserver instance for PeopleSoft portal follow instructions from “Configuring a PeopleSoft Portal” Red Paper.

After these steps are completed use `WL_HOME\config\peoplesoft\startPIA?.cmd (.sh)` to start the N’th Weblogic instance.

PeopleTools 8.44-8.48

When you install PIA and select the Multi-server domain option a domain configuration gets generated hat contains among other settings, two WebLogic instances named PIA1 and PIA2 in a WebLogic cluster to be used as a starting point for a multi server configuration.

To start these servers, in addition to the required admin server, excute the following commands from within your WebLogic domain directory.

```
startWebLogicAdmin
startManagedWebLogic PIA1
startManagedWebLogic PIA2
```

To define the same three servers as Windows services excute the following commands from within your WebLogic domain directory. When a server name is not supplied to the installNTservice script, the WebLogicAdmin server is assumed.

```
installNTservice
installNTservice PIA1
installNTservice PIA2
```

In both of these situations the WebLogicAdmin server must be fully started prior to starting PIA1, PIA2, or any managed server that is dependent on that admin server.

To define additional managed servers, such as PIA3 and PIA4, perform the following;

1. Backup the `PS_HOME/webserv/<weblogic_domain>/config.xml` file

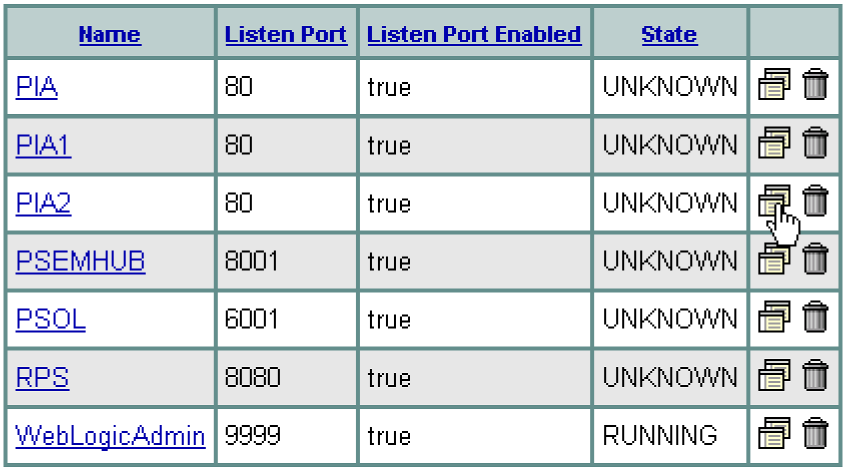
2. Start the WebLogicAdmin server by running the startWebLogicAdmin script or starting the corresponding Windows service.

3. In a browser navigate to the console at `http://webserver:9999/console`

   NOTE: The default HTTP port for the WebLogicAdmin server is 9999 and the default login is - system/password unless it was changed during the PIA install when this domain was created.

4. Select `peoplesoft->Servers` from the left frame

5. To create a new server click on the 'Configure a new Server…' link, or clone an existing server click on that clone icon in the server list. The clone icon appears as two cascading windows.



Name	Listen Port	Listen Port Enabled	State	
PIA	80	true	UNKNOWN	🗗 🗑
PIA1	80	true	UNKNOWN	🗗 🗑
PIA2	80	true	UNKNOWN	🗗 🗑
PSEMHUB	8001	true	UNKNOWN	🗗 🗑
PSOL	6001	true	UNKNOWN	🗗 🗑
RPS	8080	true	UNKNOWN	🗗 🗑
WebLogicAdmin	9999	true	RUNNING	🗗 🗑

6. Once the servers are added, configure the newly added servers to be part of the cluster (PeopleSoftCluster which comes with multiserver

PIA installation) and make sure the servers are configured to run on right listen address and listetn port.

7. You can start using the cluster, once you make sure that the servers in the cluster are all started properly. For troubleshooting errors, check the respective server logs.

Regardless of if you clone an existing server or create a new one, specify a unique server name, and IP/port combination for it. If you try to add servers from a different domain to the same cluster, make sure that you have server names unique. You can not have PIA1 and PIA2 in domain1 and PIA1 and PIA2 in domain2 and try to join all these servers in a cluster called "peoplesoftCluster". Instead you can have server names PIA3 and PIA4 in domain2 and can join these two servers in the cluster. In addition, on the 'Logging' tab for your new server(s) adjust the log file names to reference correct server name.The recommendation is to use PIA1 and PIA2 in cluster and add more servers (like PIA3, PIA4.. PIAn) instead of adding server "PIA" to the cluster.

PeopleTools 8.49 – 8.50

Starting from WebLogic 9.2 (including WebLogic Server 10.3.1), the configuration files are located in a different directory structure from previous releases of WebLogic versions.

In order to setup cluster, it is required to do a multiserver installation during PIA install. To start the multiple servers, in addition to the required admin server, excute the following commands from within your WebLogic domain's *bin* directory.

```
startWebLogicAdmin

startManagedWebLogic PIA1

startManagedWebLogic PIA2
```

To define the same three servers as Windows services excute the following commands from within your WebLogic domain's *bin* directory. When a server name is not supplied to the installNTservice script, the WebLogicAdmin server is assumed.

```
installNTservice – This will install WebLogic Admin server as a windows service

installNTservice PIA1

installNTservice PIA2
```

In both of these situations the WebLogicAdmin server must be fully started prior to starting PIA1, PIA2, or any managed server that is dependent on that admin server.

To define additional managed servers, such as PIA3 and PIA4, perform the following;

1. For PT8.49, Backup the PS_HOME/webserv/<weblogic_domain>/config/config.xml file

   For PT8.50, Backup the PIA_HOME/webserv/<weblogic_domain>/config/config.xml file

2. Start the WebLogicAdmin server by running the startWebLogicAdmin script or starting the corresponding Windows service.

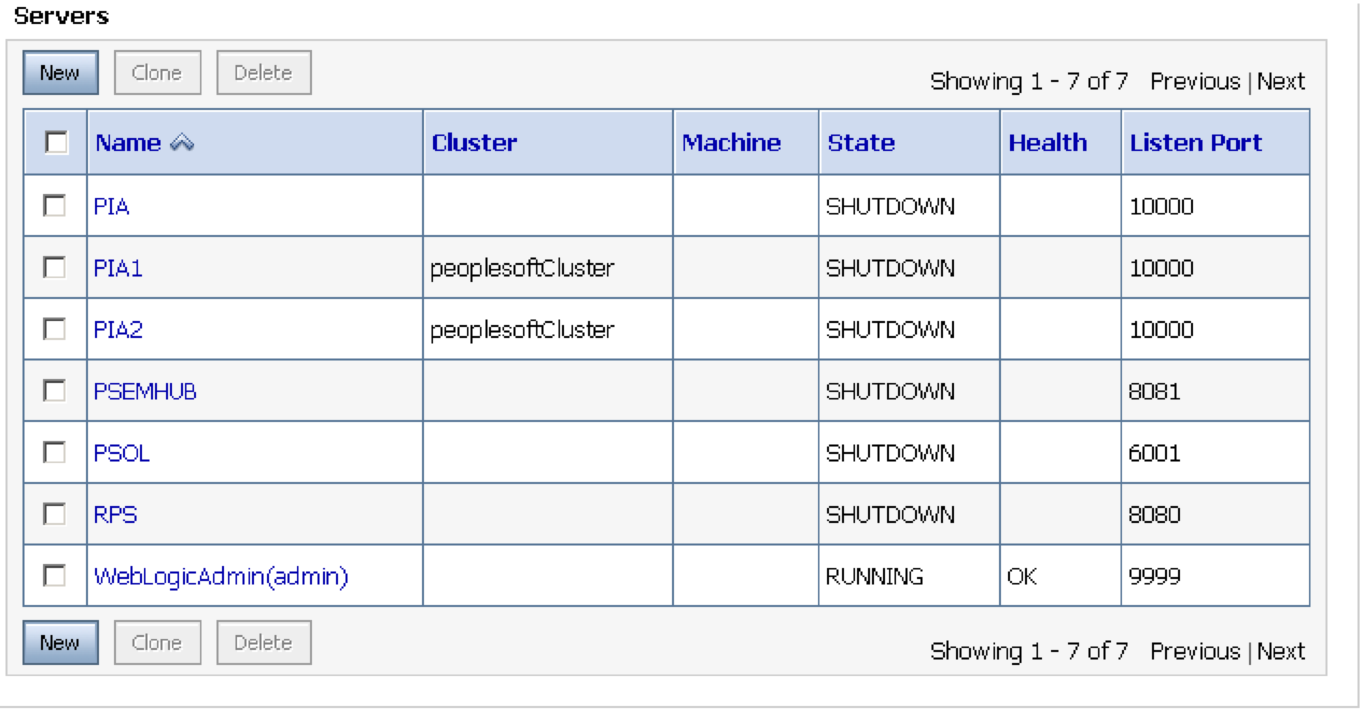
3. In a browser navigate to the console at http://webserver:9999/console

4. NOTE: The default HTTP port for the WebLogicAdmin server is 9999 and the default login is - system/password unless it was changed during the PIA install when this domain was created.

5. Select peoplesoft, Servers from the right side table

6. To create a new server click on the 'Add' button which is available on the top server list table(shown below). Add as many number of servers as per your requirement.

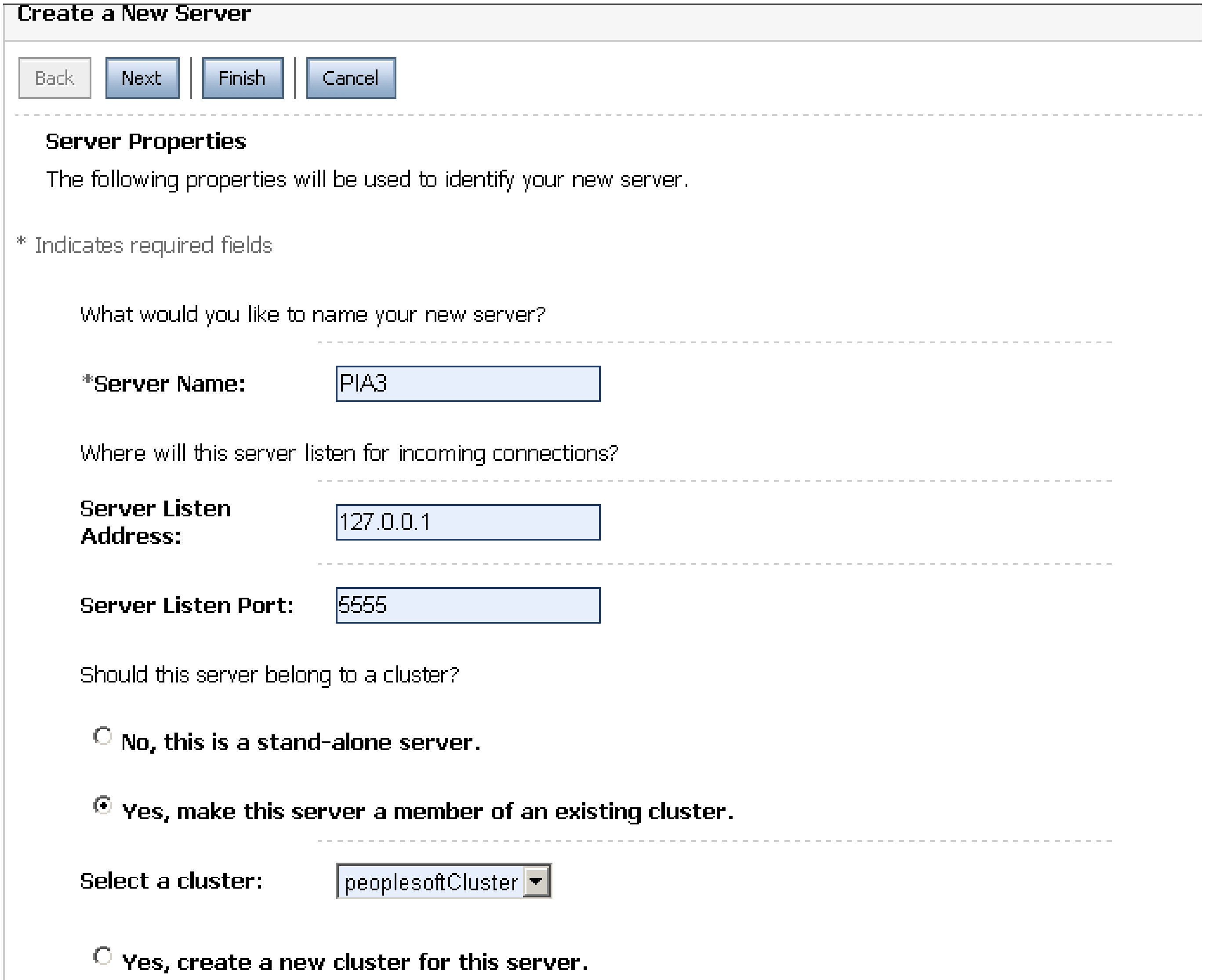
   You may also select PIA1 or PIA2, then click on "Clone" button to add additional servers.


Servers
New
Clone
Delete
Showing 1 - 7 of 7 Previous | Next

□	Name	Cluster	Machine	State	Health	Listen Port
□	PIA			SHUTDOWN		10000
□	PIA1	peoplesoftCluster		SHUTDOWN		10000
□	PIA2	peoplesoftCluster		SHUTDOWN		10000
□	PSEMHUB			SHUTDOWN		8081
□	PSOL			SHUTDOWN		6001
□	RPS			SHUTDOWN		8080
□	WebLogicAdmin(admin)			RUNNING	OK	9999

New
Clone
Delete
Showing 1 - 7 of 7 Previous | Next

7. Once the servers are created, configure them to use the right Listen Address and Port number as you specified in the web.xml and save the configuration.



Create a New Server

Server Properties

The following properties will be used to identify your new server.

* Indicates required fields

What would you like to name your new server?

*Server Name: PIA3

Where will this server listen for incoming connections?

Server Listen Address: 127.0.0.1

Server Listen Port: 5555

Should this server belong to a cluster?

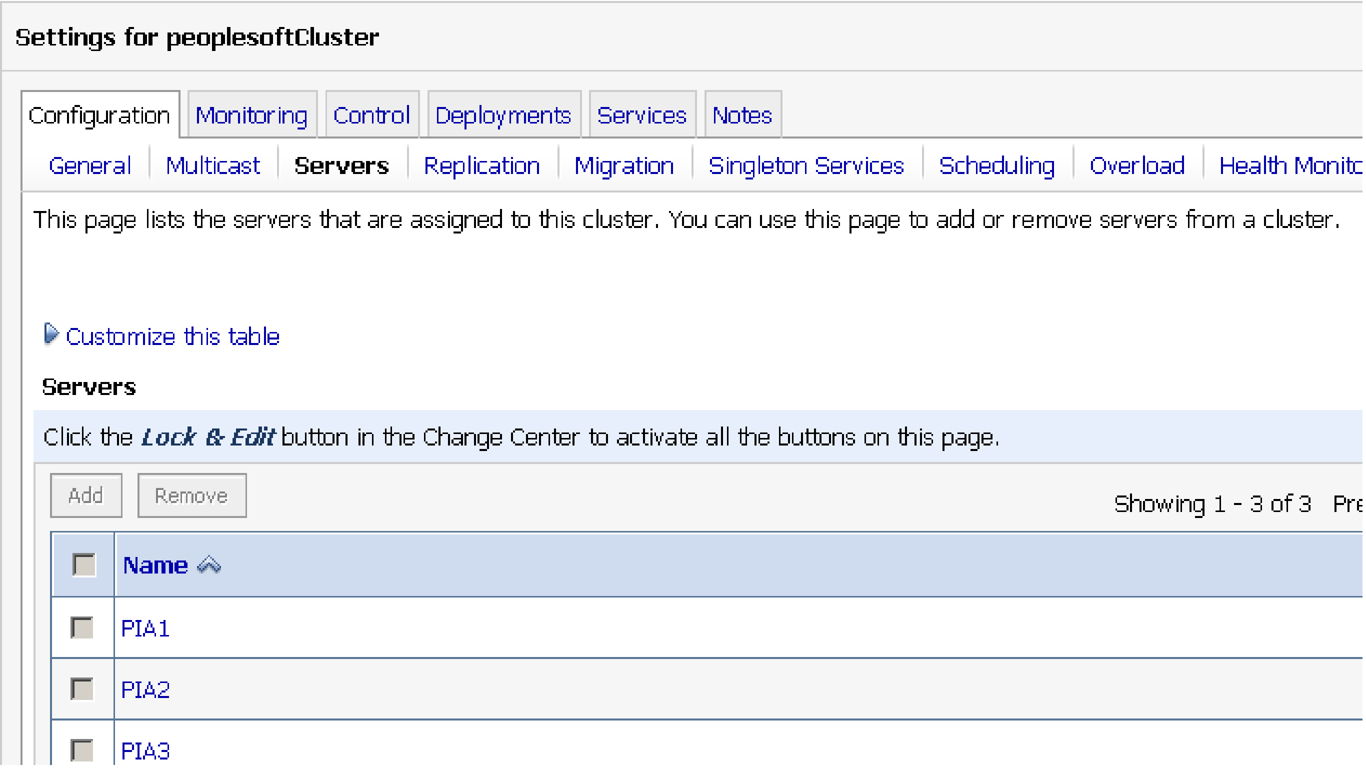
○ No, this is a stand-alone server.

● Yes, make this server a member of an existing cluster.

Select a cluster: peoplesoftCluster

○ Yes, create a new cluster for this server.

8. Go to Environment, clusters and select PeopleSoftCluster and go to “servers”. Make sure all the servers are part of the cluster.



Settings for peoplesoftCluster

This page lists the servers that are assigned to this cluster. You can use this page to add or remove servers from a cluster.

Servers

Click the ***Lock & Edit*** button in the Change Center to activate all the buttons on this page.

□	Name
□	PIA1
□	PIA2
□	PIA3

9. You can start using the cluster, once you make sure that the servers in the cluster are all started properly. For troubleshooting errors, check the respective server logs.

Regardless of if you clone an existing server or create a new one, specify a unique server name, and IP/port combination for it. If you try to add servers from a different domain to the same cluster, make sure that you have server names unique. You can not have PIA1 and PIA2 in domain1 and PIA1 and PIA2 in domain2 and try to join all these servers in a cluster called “peoplesoftCluster”. Instead you can have server names PIA3 and PIA4 in domain2 and can join these two servers in the cluster. In addition, on the ‘Logging’ tab for your new server(s) adjust the log file names to reference correct server name.The recommendation is to use PIA1 and PIA2 in cluster and add more servers (like PIA3, PIA4.. PIAn) instead of adding server “PIA” to the cluster.

How to view Reports in a clustered environment:

When you setup report distribution and nodes, make sure that all the instances of cluster point to a single shared drive and make sure that each server has access to it. Basically you have to make sure to post all the reports to a shared drive location. This way the user will be able to view the reports from any instance of cluster.

CONFIGURING A WEBSPHERE HTTP (PROXY) SERVER FOR PEOPLETOOLS 8.40-8.43

This section applies to the Simple WebSphere Clustering and Advanced WebSphere architecture. This section describes the steps to configure a *WebSphere* HTTP (proxy) server for clustering.

Installing the WebSphere Plugin

1. Install a supported HTTP server. This example will use the IBM HTTP Server V 1.3.19.
2. Stop IHS.
   - Windows
     - Go to the Services panel and stop the "IBM HTTP Server" and "IBM HTTP Administration" services.
   - Unix
     - Go to the directory where IHS is installed.
       - Solaris - `/opt/IBMHTTPD`
       - AIX - `/usr/HTTPServer`
     - Change to the bin directory
     - Execute **apachectl stop**. This will stop the IHS processes.
3. Begin the installation of WebSphere.
4. Choose custom installation.
5. Select only the **Web Server Plugins** option.
6. Select the **IBM HTTP Server** as the type of plugin to install.
7. Finish the installation of WebSphere.
8. To check that the plugin has been installed, follow the instructions below:
   - Open the **`httpd.conf`** file located in `$IHS_ROOT/conf` in a text editor.
   - Scroll to the bottom of the file. There should be several lines relating to WebSphere similar to those below (Windows example shown)

```
LoadModule ibm_app_server_http_module
C:/WebSphere/AppServer/bin/mod_ibm_app_server_h
ttp.dll
Alias /IBMWebAS/ "C:/WebSphere/AppServer/web/"
Alias /WSsamples
"C:/WebSphere/AppServer/WSsamples/"
WebSpherePluginConfig
C:\WebSphere\AppServer\config\plugin-cfg.xml
```

Configuring plugin-cfg.xml for Clustering

Assigning Server IDs and Generating Clone IDs

WebSphere Single Server uses the server's ID and the clone ID to associate requests with the appropriate
WebSphere instance.

1. Open /WebSphere/AppServer/config/server-cfg.xml for a WebSphere instance.
2. Search for the servers tag (ex. <servers xmi:type="applicationserver:ApplicationServer" ....>)
3. Within the servers tag look for the value id="-1".
4. Change the value in the quotes to a number , for example 748921. Each WebSphere instance in a cluster must have a unique server ID. The characters in the quotes must be digits or WebSphere will not start.
   <servers xmi:type="applicationserver:ApplicationServer" xmi:id="ApplicationServer_1" desiredExecutionState="START" name="Default Server" id="**748921**" moduleVisibility="APPLICATION">
5. Save and close server-cfg.xml.
6. Repeat these steps for each server-cfg.xml in the WebSphere cluster. Each WebSphere instance in a cluster must have a unique server ID. Keep track of which WebSphere instance's server ID matches with its hostname and port.

Code to build the tool to generate cloneID

Build the following java class which is the ID Generator tool:

```
public class IDGenerator {
    public static void main(String[] args) throws
java.io.IOException{
            System.out.println("Please enter the server
ID to be
translated(the server ID must consist of only numerical
digits):");

                long value = 0;
                int ch;

                while ((ch = System.in.read()) != '\n')
{
                        if (ch >= '0' && ch <= '9') {
                                value *= 10;
                                value += ch - '0';
                        }
                }

                System.out.println("The CloneID for " +
value + " in
plugin-cfg.xml is " + Long.toString(value,32));
        }
}
```

How to use the tool to generate cloneID

The server IDs need to be translated into CloneIDs that will be specified in plugin-cfg.xml. A tool can be used to generate CloneID from serverID.

1. Build the java class shown above or download the IDGenerator.class file from Customer Connection located at ftp://ftp.peoplesoft.com/outgoing/PTools/websphere/403/CloneIDGenerator/IDGenerator.class.
2. To run Clone IDGenerator, go to the directory where you have downloaded or built IDGenerator.class for example c:\temp\IDGenerator.class.
3. From command prompt run the IDGenerator as shown.
   c:\temp>{WAS_HOME}\java\bin\java IDGenerator (Windows)
   /opt/temp>{WAS_HOME}/java/bin/java IDGenerator (UNIX)
4. The program will prompt to enter a server ID.
5. Please enter the server ID set in the server-cfg.xml file.
6. Type in a server ID. For example, 748921. Only one CloneID can be generated at a time. Run the IDGenerator program once for each server ID.
7. The program will output the CloneID to be used in plugin-cfg.xml.
8. The CloneID for **748921** is mrbp
9. The CloneID **mrbp** will be used in the plugin-cfg.xml file in the next section.

Configuring plugin-cfg.xml for Clustering

1. The plugin-cfg.xml plugin configuration file (on NT C:\WebSphere\AppServer\config\plugincfg.xml) needs to be configured to communicate with multiple instances of WebSphere. A standalone HTTP (proxy) server will have the default plugin-cfg.xml plugin configuration file and is not appropriate for PeopleSoft applications. Copy over the plugin-cfg.xml plugin

configuration file from one of the WASHosts to this HTTP (proxy) host.

2. To allow the plugin to communicate correctly, a server group needs to be configured with references to the instances. Each instance needs a unique CloneID. Here is the definition of the CloneID from the IBM WebSphere V4.0 Advanced Edition Scalability redpiece.

   CloneID - Used in conjunction with Session Affinity. When this attribute is set the plug-in will check the incoming cookie header or URL for JSESSIONID. If the JSESSIONID is found then the plug-in will look for a CloneID or CloneIDs. If CloneIDs are found and a match is made to this attribute then the request will be sent to this server rather than being load balanced across the server group.
   Default Value: none
   Expected Values: A character string

3. Each WebSphere instance in the cluster must be added to the ServerGroup tag. The Server tag can be copied and edited for each WebSphere instance. The host name for each server must be specified as the machine name or IP address where the WebSphere instance(s) is running. The default hostname will be localhost. Replace localhost with the hostname of the WebSphere server and create new Server tags as necessary.

   ```
   <Server Name="Default Server">
      <Transport Hostname="WASHost1" Port="9080"
   Protocol="http"/>
   </Server>
   <Server Name="Default Server">
     <Transport Hostname="WASHost2" Port="9080"
   Protocol="http"/>
   </Server>
   ```

4. Add the CloneID for each server. Make sure the CloneID matches with the hostname and port of the ServerID used to generate the CloneID.

   ```
   <Server CloneID="mrbp" Name="Default Server">
      <Transport Hostname="WASHost1" Port="9080"
   Protocol="http"/>
   </Server>
   <Server CloneID="213a" Name="Default Server">
      <Transport Hostname="WASHost2" Port="9080"
   Protocol="http"/>
   </Server>
   ```

Below is a sample `plugin-cfg.xml` file. The changes are in bold. (Hint: the <Server> tag can be copied and pasted from the first entry, and the CloneID, hostname and port changes can be altered.)

```
<?xml version="1.0"?>
<Config>
    <Log LogLevel="Error"
Name="C:\WebSphere\AppServerSingle/logs\native.log"/
>
    <ServerGroup Name="Single_Server_Group">
        <Server CloneID="mrbp" Name="Default
Server">
            <Transport Hostname="WASHost1"
Port="9080" Protocol="http"/>
        </Server>
      <Server CloneID="mrbq" Name="Default Server">
         <Transport Hostname="WASHost1" Port="9081"
Protocol="http"/>
      </Server>
        <Server CloneID="mrbr" Name="Default
Server">
          <Transport Hostname="WASHost2" Port="9080"
Protocol="http"/>
      </Server>
        <Server CloneID="mrbs" Name="Default
Server">
          <Transport Hostname="WASHost2" Port="9081"
Protocol="http"/>
      </Server>
    </ServerGroup>
    <VirtualHostGroup Name="default_host">
        <VirtualHost Name="*:80"/>
        <VirtualHost Name="*:443"/>
    </VirtualHostGroup>
    <VirtualHostGroup Name="admin_host">
        <VirtualHost Name="*:9090"/>
    </VirtualHostGroup>
    <UriGroup Name="default_host_URIs">
      <Uri Name="/servlet/*"/>
      <Uri Name="/psc/*"/>
      <Uri Name="/psp/*"/>
      <Uri Name="/cs/*"/>
      <Uri Name="/xmllink/*"/>
      <Uri Name="/psreports/*"/>
      <Uri Name="/SchedulerTransfer/*"/>
      <Uri Name="/SyncServer/*"/>
      <Uri Name="/PSIGW"/>
      <Uri Name="/PSINTERLINKS"/>
      <Uri Name="/ps/*"/>
      <Uri Name="/psftmodified/*"/>
      <Uri Name="/webapp/examples"/>
      <Uri Name="*.jsp"/>
      <Uri Name="/ErrorReporter"/>
      <Uri Name="/j_security_check"/>
      <Uri Name="/tradetheme"/>
      <Uri Name="/theme"/>
```

```
            <Uri Name="/WebSphereSamples/*"/>
        </UriGroup>
        <Route ServerGroup="Single_Server_Group"
            UriGroup="default_host_URIs"
    VirtualHostGroup="default_host"/>
        <UriGroup Name="admin_host_URIs">
            <Uri Name="/admin/*"/>
        </UriGroup>
        <Route ServerGroup="Single_Server_Group"
    UriGroup="admin_host_URIs"
    VirtualHostGroup="admin_host"/>
</Config>
```

CONFIGURING MULTIPLE INSTANCES OF WEBSPHERE SINGLE SERVER FOR PEOPLETOOLS 8.40-8.43

This section applies to all Webserver Clustering architecture that uses WebSphere. Additionally, if you are configuring a SimpleWebSphere Cluster or an Advanced WebSphere cluster then you will also need to configure an HTTP (proxy) server as described in the previous section. First, follow PeopleSoft Internet Architecture (PIA) installation instructions to configure and test WebSphere on the webserver host. The instructions below assumes that WebSphere is up and running on the host.

1. Copy `$WAS_ROOT/config/server-cfg.xml` to `server-cfg2.xml`. In the table below, Instance 1 will relate to the original `server-cfg.xml`, and Instance 2 will relate to `server-cfg2.xml`

2. Open `server-cfg2.xml` in a text editor.

3. The following entries need to be changed.

Port Use	Instance 1	Instance 2	XML tag
OLT	2102	2103	<objectLevelTraceSettings xmi:id="ObjectLevelTrace_1" enable="false" hostname="localhost" port="***2102***" debug="false" sourcePath=""/>
LSD	9000	9001	<locationServiceDaemon xmi:id="LocationServiceDaemon_1" hostname="localhost" port="**9000**" mode="NONE"/>
Administrative Debugging	7000	7001	<traceService xmi:id="TraceServiceConfig_1" enable="true"

			traceSpecification="*=all=disabled" traceOutputFilename="stdout" diagThreadPort="***7000***"/>
Bootstrap	900	901	<orbSettings xmi:id="ORBConfig_1" enable="true" bootstrapHost="localhost" bootstrapPort="***900***">
HTTP Transport Port	9080	9081	<transports xmi:type="applicationserver:HTTPTransport" xmi:id="HttpTransport_1" hostname="*" port="***9080***"> </transports>
HTTPS Transport Port	9443	9444	<transports xmi:type="applicationserver:HTTPTransport" xmi:id="HttpTransport_2" hostname="*" port="***9443***" sslEnabled="true"> </transports>
Administrative Console	9090	9091	<transports xmi:type="applicationserver:HTTPTransport" xmi:id="HttpTransport_3" hostname="*" port="***9090***" external="false"> </transports>

Note: These ports must be available on the machine. Please use netstat or a similar tool to check for port availability. These port values can be changed to any open port. If WebSphere will run as a non-root user on a UNIX system, the bootstrap port must be > 1024.

8. Change the virtual host entry for the http/https transports. Look for these tags and change the highlighted values to 9081 and 9444.
<virtualHosts xmi:id="VirtualHost_1" name="default_host">
<aliases xmi:id="HostAlias_1" hostname="*" port="***9080***"/>
<aliases xmi:id="HostAlias_2" hostname="*" port="***9443***"/>

4. Change directory and file names

File or Directory Name	Instance 1	Instance 2	XML tag
Passivation Directory	${WAS_ROOT}/temp	${WAS_ROOT}/temp2	<ejbContainer xmi:id="EJBContainer_1" passivationDirectory="***${WAS_ROOT}/temp***" inactivePoolCleanupInterval="30000" installedEJBModules="EJBModuleRef_2 EJBModuleRef_1 EJBModuleRef_3

			EJBModuleRef_4 EJBModuleRef_5 EJBModuleRef_6 EJBModuleRef_7 EJBModuleRef_8 EJBModuleRef_9 EJBModuleRef_10 EJBModuleRef_11 EJBModuleRef_12" defaultDatasource="DataSource_1">
LOG_ROOT	${WAS_ROOT}/logs	${WAS_ROOT}/logs2	<entries xmi:id="PathMapEntry_2" symbolicName="LOG_ROOT" path="***${WAS_ROOT}/logs***" description="The filesystem path to the directory which will contain server log files."/>
TRANLOG_ROOT	${WAS_ROOT}/tranlog	${WAS_ROOT}/tranlog2	<entries xmi:id="PathMapEntry_3" symbolicName="TRANLOG_ROOT" path="**${WAS_ROOT}/tranlog**" description="The filesystem path to the directory which will transaction log files."/>

These settings assume that:

- The same key rings are being used for security on each server. If not, replace the key file and the trust file references with the appropriate files.
- The same installed application directory is shared between the instances.
- The instances are using the same security settings.

10. The directories created in the previous table must be manually created. Change to **$WAS_ROOT** and create the directories (temp2, logs2, tranlog2).

11. Each instance may be started using a modified start script. Change to the **$WAS_ROOT/bin** directory.
    Execute **startServer.bat** to start the first instance.
    Execute **startServer.bat -configFile $WAS_ROOT/config/server-cfg2.xml** to start the second instance.

12. Each instance must be stopped with a modified stop script. Change to **$WAS_ROOT/bin**.
    Execute **stopServer.bat** to stop the first instance.
    Execute **stopServer.bat -configFile $WAS_ROOT/config/server-cfg2.xml** to stop the second instance.

13. The above steps walk through configuring two WebSphere instances on a single machine. Use this process to create more than two instances by simply incrementing the port numbers by one. Please use **netstat** or a similar tool to avoid using ports in use by other processes.

    The limit on the number of instances is the amount of system resources available. JVMs should be able to keep memory in system memory instead of using swap space. Creating more than 10 instances will require changing the port numbering convention because the HTTP transport and the Administrative console will overlap at that point. Any open ports may be used.

10. **Optional:** If a separate DNS name is configured for the proxy/loadbalancer or an SSL accelerator is used on those devices, it will be necessary to setup **`pswebservername`**, **`defaultScheme`** and **`defaultPort`** in those cases.

Optional: To configure the webserver instance for PeopleSoft portal follow instructions from “ Configuring a PeopleSoft Portal” Red Paper.

CONFIGURING A WEBSPHERE HTTP (PROXY) SERVER WITH PEOPLETOOLS 8.44-8.48

The installation of the WebSphere plugin into the HTTP Servers is documented in the WebSphere Installation Guide. The WebSphere plugin uses a configuration file (plugin-cfg.xml) to route requests to WebSphere JVMs. This file needs to be updated whenever the PeopleSoft Internet Architecture (PIA) is installed or uninstalled. The updated file should then be copied to the HTTP Server machine to route requests appropriately. The steps for each HTTP server are below.

IBM Http Server (IHS) Plugin

If you are using IHS as the remote Http server, follow the instructions in the WebSphere Installation Guide to install the WebSphere plugin into IHS. Refer to the “Working with WebSphere” section of the “PeopleTools: Server Tools” PeopleBook for more information on configuring the plugin. Once installation is complete and has been tested with a WebSphere instance, follow these steps to allow clustering with IHS and WebSphere.

Configuring IHS with plugin-cfg.xml

1. If IHS is on a separate machine than WebSphere, the plugin-cfg.xml file will need to be copied to the machine. If IHS is on the same machine, the plugin will only need to be regenerated (execute this step only). In a browser from the IHS machine, go to the WebSphere Admin Console (http://<hostname>:9090/admin by default):

    a. Select Environment->Update Web Server Plugin.

    b. Click "OK".

2. To copy the plugin-cfg.xml file to the IHS machine, right click on "View or download the current web server plugin configuration file" and "Save Target As…". Save the file to a temporary directory on the IHS machine.

3. To find the location of the plugin-cfg.xml file, open IBM_HTTP_Server_HOME/conf/httpd.conf. Search for "plugin-cfg.xml" to find its location on the IHS machine.

4. Copy the plugin-cfg.xml file in the temporary directory to the location in step 3.

SunOne (formerly iPlanet) Plugin

If you are using SunOne as the remote Http server, follow the instructions in the WebSphere Installation Guide to install the WebSphere plugin into SunOne. Refer to the "Working with WebSphere" section of the "PeopleTools: Server Tools" PeopleBook for more information on configuring the plugin. Once installation is complete and has been tested with a WebSphere instance, follow these steps to allow clustering with SunOne and WebSphere.

Configuring SunOne with plugin-cfg.xml

1. If SunOne is on a separate machine than WebSphere, the plugin-cfg.xml file will need to be copied to the machine. If SunOne is on the same machine, the plugin will only need to be regenerated (execute this step only). In a browser from the SunOne machine, go to the WebSphere Admin Console (http://<hostname>:9090/admin by default).

    a. Select Environment->Update Web Server Plugin.

    b. Click "OK".

2. To copy the plugin-cfg.xml file to the SunOne machine, right click on "View or download the current web server plugin configuration file" and

"Save Target As…". Save the file to a temporary directory on the SunOne machine.

- To find the location of the plugin-cfg.xml file, open Sun_ONE_HOME/servers/https-<machine>.domain>/config/magnus.conf ,search for plugin-cfg.xml file in it to find out the location of plugin-cfg.xml file on the SunOne machine.
- Copy the plugin-cfg.xml in the temporary directory to the location in step 3.

Microsoft Internet Information Server (IIS) Plugin

If you are using IIS as the remote Http server, follow the instructions in the WebSphere Installation Guide to install the WebSphere plugin into IIS. Refer to the "Working with WebSphere" section of the "PeopleTools: Server Tools" PeopleBook for more information on configuring the plugin. Once installation is complete and has been tested with a WebSphere instance, follow these steps to allow clustering with IIS and WebSphere.

Configuring IIS with plugin-cfg.xml

1. If IIS is on a separate machine than WebSphere, the plugin-cfg.xml file will need to be copied to the machine. If IIS is on the same machine, the plugin will only need to be regenerated (execute this step only). In a browser from the SunOne machine, go to the WebSphere Admin Console (http://<hostname>:9090/admin by default).
    a. Select Environment->Update Web Server Plugin.
    b. Click "OK".
2. To copy the plugin-cfg.xml file to the IIS machine, right click on "View or download the current web server plugin configuration file" and "Save Target As…". Save the file to a temporary directory on the IIS machine.
3. To find the location of the plugin-cfg.xml file, Open the Windows registry, expand HKEY_LOCAL_MACHINE > SOFTWARE > IBM > WebSphere Application Server > 5.0.0.0.Plugin Config. The key points to the plugin-cfg.xml file on the RPS machine
4. Copy the plugin-cfg.xml in the temporary directory to the location in step 3

CONFIGURING A WEBSPHERE HTTP (PROXY) SERVER WITH PEOPLETOOLS 8.49

The installation of the WebSphere plugin into the HTTP Servers is documented in the WebSphere installation instructions section of the PT 8.49 install guide.

IBM Http Server (IHS) Plugin

If you are using IHS as the remote Http server, follow the instructions in the WebSphere 6.1.0.3 Installation Guide to install the WebSphere plugin into IHS. Refer to the “Working with WebSphere” section of the “PeopleTools: Server Administration guide” PeopleBooks for more information on configuring the plugin. Once installation is complete and has been tested with a WebSphere instance, follow these steps to allow clustering with IHS and WebSphere. The following steps assume that you configured the plugin to use with IHS.

Configuring IHS with plugin-cfg.xml

1. If IHS is installed on a separate machine than WebSphere, the plugin-cfg.xml file will need to be copied to the machine. If IHS is on the same machine, the plugin will only need to be regenerated (execute this step only).

    - Copy IHS_HOME\Plugins\bin\configurewebserver1.bat to WAS_HOME\bin
    - change directory to WAS_HOME\bin
    - Run configurewebserver1.bat or .sh
    - This will create a plugin-cfg.xml under IHS_HOME\Plugins\config\webserver1

2. If IHS is running on a separate machine than WebSphere, copy the plugin-cfg.xml to IHS machine.
3. To find the location of the plugin-cfg.xml file, open IBM_HTTP_Server_HOME/conf/httpd.conf. Search for “plugin-cfg.xml” to find its location on the IHS machine
4. Copy the plugin-cfg.xml in the temporary directory to the location in step 3

Configuring IIS and Sun One

If you are using IIS or SunONE as the remote HTTP Serve, follow the steps mentioned in the IBM’s Info Center to configure the plug-ins

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.wsfep.multiplatform.doc/info/ae/ae/tins_manualWebIIS.html

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tins_manualWebIPL.html

Once the plug-in is configured, you need to generate the plugin-cfg.xml file using and copy this file to an appropriate location based on the IIS or SunOne install.

You can find the location of the existing plugin-cfg.xml on the proxy machine as follows:

- On an IBM Http Server machine —
  - Open IBM_HTTP_Server_HOME\conf\httpd.conf, and search for "plugin-cfg.xml" in it to discover the location of the plugin-cfg.xml file on the proxy machine.
- On a Sun ONE (iPlanet) Web Server machine —
  - Open Sun_ONE_HOME\servers\https-machine.domain\config\magnus.conf, and search for "plugin-cfg.xml" in it to discover the location of the plugin-cfg.xml file on the proxy machine.
- On a Microsoft IIS machine —
  - Open the Windows registry, expand HKEY_LOCAL_MACHINE > SOFTWARE > IBM > WebSphere Application Server > 5.0.0.0.Plugin Config. The key points to the plugin-cfg.xml file on the proxy machine

CONFIGURING A WEBSPHERE CLUSTER WITH PEOPLETOOLS 8.44-8.48

WebSphere Terms

WebSphere includes two product components with PeopleTools 8.44 instead of the single component included in previous PeopleTools releases.

Products

- IBM WebSphere Application Server V5.1 Base – a J2EE V1.3 compliant java application server. Provides a java virtual machine (JVM) for the PIA to run within. Installs application servers and node agent.

- IBM WebSphere Application Server Network Deployment (ND) V5.1 – provides central administration of multiple WebSphere Application Server Base processes on one or more machines. Installs deployment manager.

WebSphere has different components that provide different functions.

Application Server – provides a java runtime environment for the PIA.

Node – consists of a set of application servers running on a single machine.

Node Agent – process that runs on a node to manage the machine's application servers.

Cell – a grouping of nodes into a single administrative domain.

Deployment Manager (Cell Manager) – a java process that controls and communicates with all node agents within a cell.

Cluster – represents a group of cooperating application servers to create workload managed environment. Each application server runs the same enterprise application (PIA).

Cluster Member – an application server that is part of a cluster.

Components

WebSphere ND uses the Deployment Manager to manage one or more WebSphere application server nodes. Each node can have one or more WebSphere application servers. There are two nodes and two application servers on each node in this diagram.

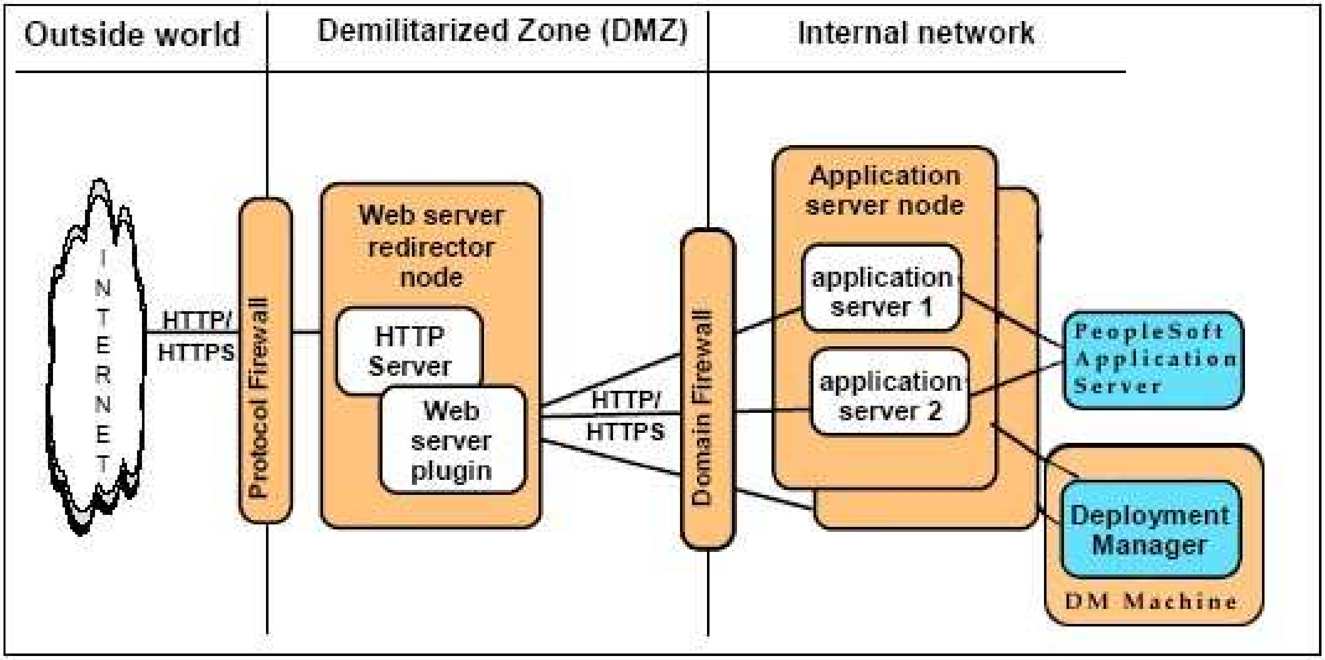


Figure 1 – WebSphere ND components

Runtime Architecture

WebSphere components can be installed and configured on multiple machines. Figure 2 shows a sample three-tier architecture, with the HTTP server, WebSphere Application Server and the Deployment Manager running on separate machines.

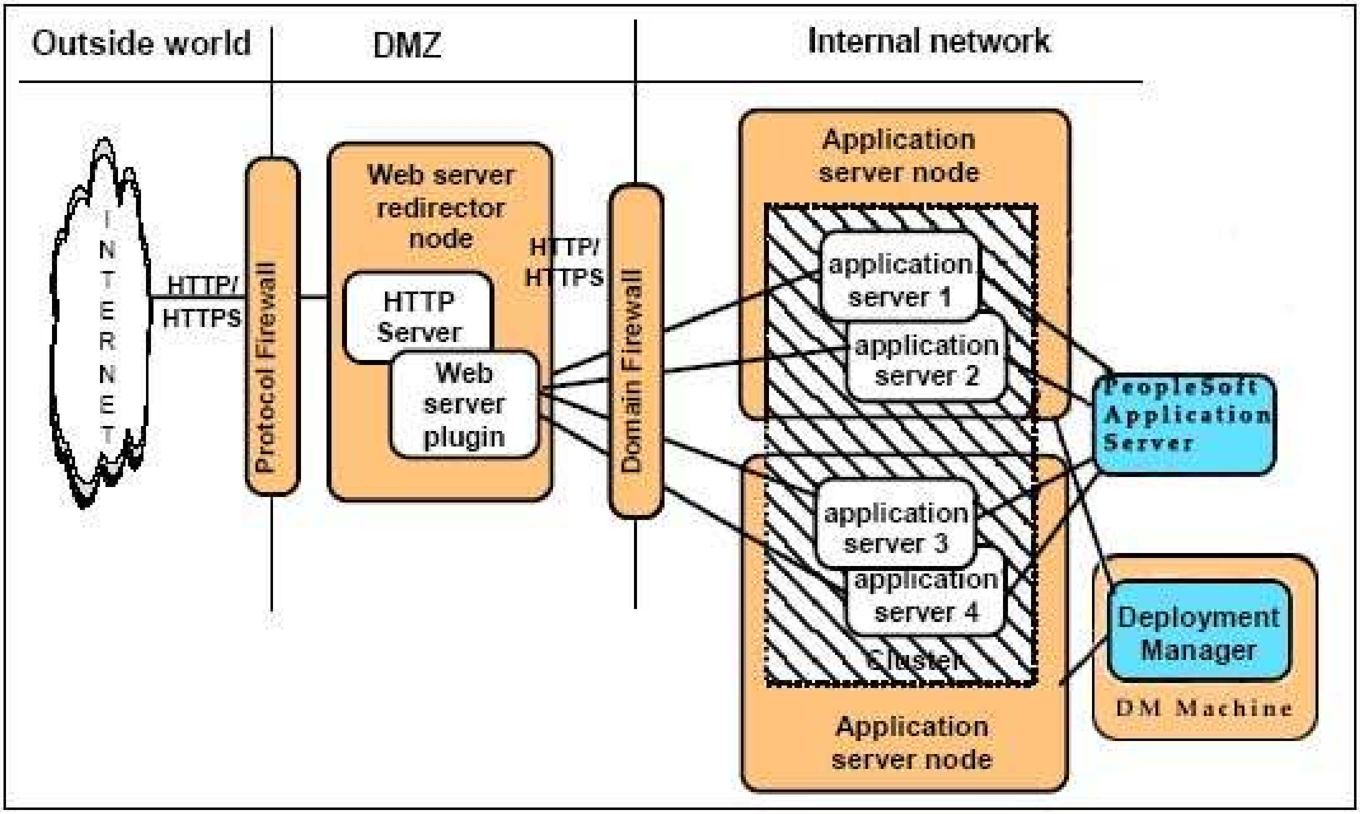


Figure 2 – Sample 3-machine architecture

Federating a Node

Each WebSphere Base node can be federated to WebSphere ND i.e to a cell using WebSphere Base's addNode command.

Assumptions

This file describes the steps necessary to install the PIA on one or more servers, which will be clustered together.

<WAS_HOME> is the directory where WebSphere is installed. The defaults are:

Operating System	**Default install directory**
AIX	Base: /usr/WebSphere51/AppServer ND: /usr/WebSphere51/DeployMgr
HP-UX	Base: /opt/WebSphere51/AppServer ND: /opt/WebSphere51/DeployMgr

Linux	Base: /opt/WebSphere51/AppServer ND: /opt/WebSphere51/DeployMgr
Solaris	Base: /opt/WebSphere51/AppServer ND: /opt/WebSphere51/DeployMgr
Windows	Base: C:\WebSphere51\AppServer ND: C:\WebSphere51\DeployMgr

WebSphere Clustering Process Overview

The PIA deploys only to a WebSphere Base instance. To create a WebSphere cluster, the PIA must first be deployed to a WebSphere Base instance and then customized (adding additional PeopleSoft sites, html customizations, etc.). The PIA deployed to the WebSphere Base instance will then be exported as an ear file. The ear file can then be deployed to a WebSphere ND cluster. **It is strongly recommended that the initial WebSphere Base instance be used as a staging server to update with patches and customizations to the PIA**. Updates can be exported as an ear file to be deployed to the cluster.

NOTE: The staging server should be on the same operating system as the production environment. The hard coded directory links in the PIA require that the PIA ear be installed to the same directory in each environment.

There are some extra steps required to deploy the PIA to a WebSphere cluster. Here is a high level overview of the clustering process. The step-by-step process follows in the next section.

1. Install WebSphere Base & ND software.
2. Install the PIA into a single WebSphere Base instance.
3. Create a single PeopleSoft ear file from the installed PIA.
4. Add nodes to the cell.
5. Create a cluster.
6. Create cluster members.
7. Deploy the PeopleSoft ear file to the cluster.
8. Make any necessary changes to the virtual hosts and the plugin-cfg.xml.

9. Start the cluster.

Configuring a WebSphere Cluster

1. Install WebSphere Deployment Manager on the *DeploymentManager* machine. Use the instructions provided in the PeopleTools Installation Documents.
2. Install WebSphere Base on other servers (*WebSphere1, WebSphere2…WebSphereN*). Use the instructions provided in the PeopleTools Installation Documents.
3. If using an HTTP Server, install a supported HTTP Server on *HTTPServer* machine. Use the instructions provided in the PeopleTools Installation Documents.
4. On *WebSphere1*, install the PIA, record settings, and create the Ear file.
    a. Install PIA using the instructions in the PeopleTools Installation Documentation

    b. Record PS_HOME, AuthTokenDomain, application name and psreports path. This information is very important and should be saved for future reference. Note: **The value for PS_HOME will be used by all machines in the cluster, so make sure the path is valid for each machine.**

    c. Create any additional PeopleSoft sites.

    d. Perform any required customizations.

    e. Test to ensure setup works.

    f. Execute WebSphere's EARExpander from WAS_HOME/bin to create an EAR file with the customizations. The operationDir value should be /PS_HOME/webserv/<cell_name>_<node_name>_<servername>/<application name>.
    Windows

    ```
    C:\<WAS_HOME>\bin\EARExpander.bat –ear
    C:\PS_HOME\PeopleSoft.ear –operationDir
    C:\PS_HOME\webserv\dbarona30node_dbarona30node_
    server1\peoplesoft.ear –operation collapse
    ```

    Unix
    /<WAS_HOME>/bin/EARExpander.sh –ear
    /PS_HOME/PeopleSoft.ear –operationDir
    /PS_HOME/webserv/DeployedPeopleSoft.ear –operation
    collapse

5. Be sure the *DeploymentManager* is started.
   Windows
   `C:\WebSphere\DeploymentManager\bin\startManager.bat`
   Unix
   `/WebSphere/DeploymentManager/bin/startManager.sh`

6. On *WebSphere1* to *WebSphereN*

   a. Before federating the node, check that the date and times for the machines are within 5 minutes of each other. If the machines do not have similar times, an error message will appear and the node will not be added.

   b. The Deployment Manager machine and the Node Agent machine must be able to resolve the hostnames of each other. To test, use the hostname with the ping command.

   c. Add the node to the server configuration using the command:
      Windows

      ```
      C:\WebSphere51\AppServer\bin\addNode.bat
      <DeploymentManager> <8879>
      ```

      Unix
      ```
      /WebSphere51/AppServer/bin/addNode.sh
      <DeploymentManager> <8879>
      ```

      where <DeploymentManager> is the DeploymentManager hostname and <8879> is the SOAP connector port. This will stop any servers currently running on the node. **Note the command does *not* use the `–includeapps` parameter.**

      1. The Soap Connector Port is listed in the serverindex.xml file, which is located at:
         Windows
         C:\WebSphere51\DeploymentManager\config\cells\<Network>\nodes\<NodeManager>\serverindex.html
         Unix
         /WebSphere51/DeploymentManager/config/cells/<Network>/nodes/<NodesManager>/serverindex.html

      2. Open serverindex.html and look for the Soap tag:

```
        <specialEndpoints xmi:id="NamedEndPoint_4"
endPointName="SOAP_CONNECTOR_ADDRESS">

          <endPoint port="8879" host="dbarona30" xmi:id="EndPoint_4"/>

        </specialEndpoints>
```

3. In the above example, the Soap port is 8879

d. The NodeManager will be started as part of the addnode command. If not, start the NodeManager using:
Windows
`C:\WebSphere51\AppServer\bin\startNode.bat`
Unix
`/WebSphere51/AppServer/bin/startNode.sh`

7. Open a browser and access the WebSphere Administrative Console *pointing to the DeploymentManager* machine e.g. http://<DeploymentManager>:9090/admin.

a. Create an *Application Server* that will be used as a template.

i. Select *Servers -> Application Servers*.

ii. Click *New.*

iii. Enter *PIAServer* as the server name.

iv. Select the option "Existing Application Server" template

v. Click *Next.*

vi. Click *Finish*.

vii. *Save* the changes.

b. Select *Servers -> Application Servers -> PIAServer* to make configuration changes.

viii. Add a shared library

1. Go to Environment->Shared Libraries.

2. Select the Server level scope and select the server where you want to define a shared library.

3. Click New button.

4. Enter a name (peoplesoft) for the library.

5. Enter classpath as given below (this example is shown where PIA is deployed under e:\pt848). Make sure you enter these paths one after another.

a. E:\pt848\webserv\LAGOURAB-PC_LAGOURAB-PC_server1\peoplesoft.ear\lib\pluto-1.0.1.jar;

b. E:\pt848\webserv\LAGOURAB-PC_LAGOURAB-PC_server1\peoplesoft.ear\lib\portlet-api-1.0.jar;

c. E:\pt848\webserv\LAGOURAB-PC_LAGOURAB-PC_server1\peoplesoft.ear\lib\saaj.jar;

d. E:\pt848\webserv\LAGOURAB-PC_LAGOURAB-PC_server1\peoplesoft.ear\lib\xalan.jar;

e. E:\pt848\webserv\LAGOURAB-PC_LAGOURAB-PC_server1\peoplesoft.ear\PSIGW\WEB-INF\lib\mail.jar;

f. E:\pt848\webserv\LAGOURAB-PC_LAGOURAB-PC_server1\peoplesoft.ear\PSIGW\WEB-INF\lib\activation.jar;

6. click ok and save.

7. Now go to Application Servers->Select PIAServer.

8. Click on Classloader.

9. and you will see a list of classloader and select the available one.

10. Click on Libraries from Additional Properties.

11. Click Add button and you will see the library "peoplesoft" in the drop down list.

12. Click Apply button and Save.

ix. Change JVM settings to the cluster members.

1. Select *Servers -> Application Servers*.

2. Select *PIAServer.*

3. Select *Process Definition*.

4. Select *Java Virtual Machine*.

5. Add the required jar files to the *Boot Classpath.*

6. Add these values to the *Boot Classpath*.
`PS_HOME/webserv/<WebSphereNodeDirectory>/peoplesoft.ear/PSIGW/WEB-`

`INF/lib/activation.jar;PS_HOME/webserv/<cell_name>_<node_name>_<application_server>/peoplesoft.ear/PSIGW/WEB-INF/lib/mail.jar;`

For Example:
`C:/PS_HOME/webserv/RSHANKA2040303Node_RSHANKA2040303Node_server1/peoplesoft.ear/PSIGW/WEB-INF/lib/activation.jar;C:/PS_HOME/webserv/RSHANKA2040303Node_RSHANKA2040303Node_server1/peoplesoft.ear/PSIGW/WEB-INF/lib/mail.jar;`

7. Click *Apply*.

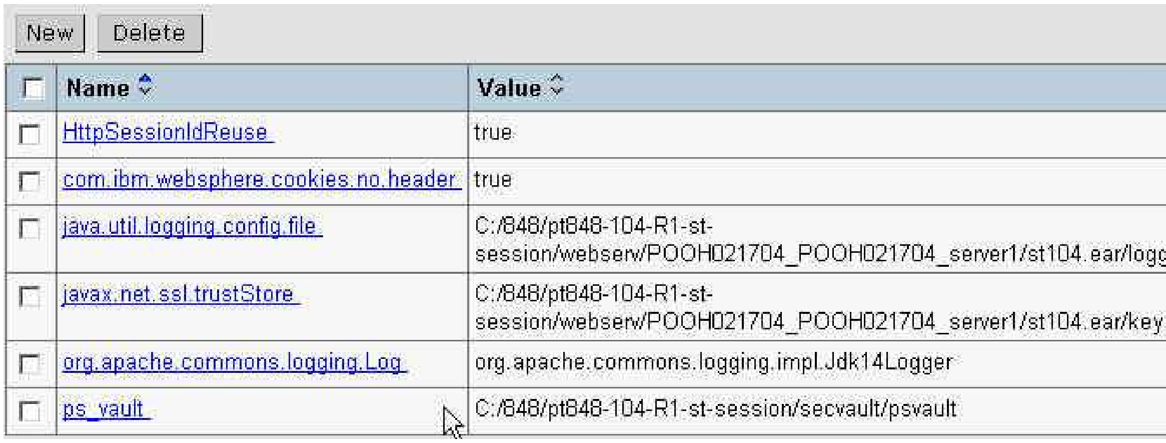
x. Create custom properties under *Java Virtual Machine*. It is a good practice check the base install and verify you have all these custom properties defined under server1's Java Virtual Machine.

1. Scroll to the bottom of the page and select *Custom Properties*.
2. Click *New*.
3. For the name, enter: `com.ibm.websphere.cookies.no.header`
4. For the value, enter: true
5. Click *OK*.
6. Repeat steps 1 to 5 and enter a new property called: `HttpSessionIdReuse and value: true`. **Note: Starting from PT 8.47.15 and PT 8.48.13, you need to set this value to false.**
7. Repeat steps 1 to 5 and enter a new property called : `javax.net.ssl.trustStore` and set value to `PS_HOME/webserv/<cell_name>_<node_name>_<application_server>/peoplesoft.ear/keystore/pskey`
8. Repeat steps 1 to 5 and enter a new property called : `java.util.logging.config.file` and set value to `PS_HOME/webserv/<cell_name>_<node_name>_<application_server>/peoplesoft.ear/logging.properties`

9. Repeat steps 1 to 5 and enter a new property called : `org.apache.commons.logging.Log` and set value to `org.apache.commons.logging.impl.Jdk14Logger`

10. Repeat steps 1 to 5 and enter a new property called : `ps_vault` and set value to `PS_HOME`/secvault/psvault or `PS_HOME/webserv/<cell_name>_<node_name>_<application_server>/peoplesoft.ear/psvault (based on the tools release)`

`You should see the following custom properties under Java Virtual Machine.`



□	Name	Value
□	HttpSessionIdReuse	true
□	com.ibm.websphere.cookies.no.header	true
□	java.util.logging.config.file	C:/848/pt848-104-R1-st-session/webserv/POOH021704_POOH021704_server1/st104.ear/logg
□	javax.net.ssl.trustStore	C:/848/pt848-104-R1-st-session/webserv/POOH021704_POOH021704_server1/st104.ear/key
□	org.apache.commons.logging.Log	org.apache.commons.logging.impl.Jdk14Logger
□	ps_vault	C:/848/pt848-104-R1-st-session/secvault/psvault

xi. Add the AuthTokenDomain to the session management.

1. Click *Application Server -> PIAServer.*

2. Click *Web Container.*

3. Click *Session Management*.

4. Click *Enable Cookies.*

5. Add the *AuthTokenDomain* value specified during the PIA installation in step 4b in the *Cookie Domain* textbox.

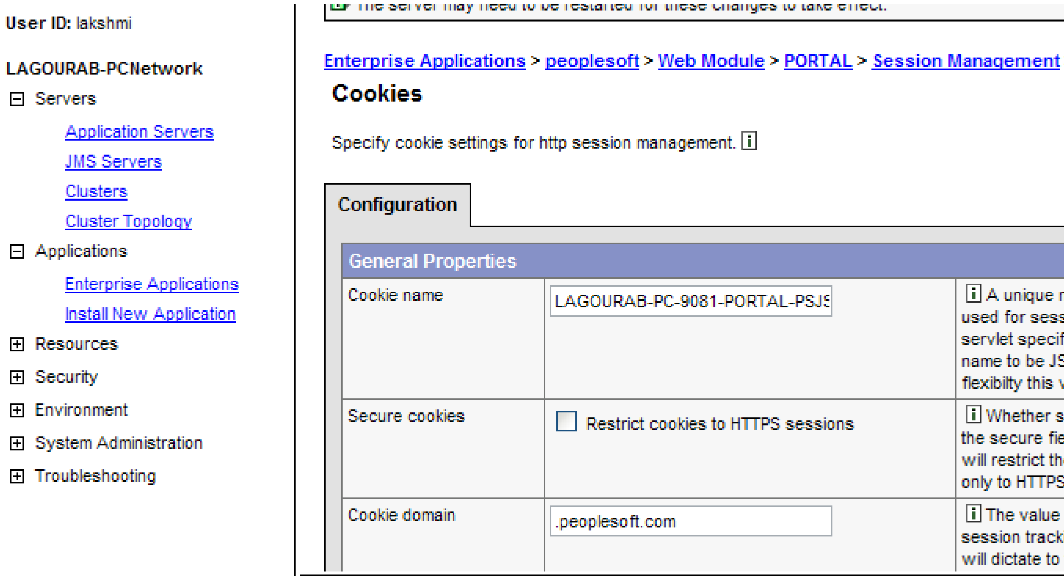
6. Click *OK*.

7. Save the changes.

8. Starting from PT 8.47.15 and PT 8.48.13, JSESSIONID sshould be renamed as a result of Session Fixation feature. Please follow the steps listed below to rename the JSESSIONID.

1. Go to WAS ND admin console – http://host:port/admin
2. Go to Enterprise Applications
3. Click on peoplesoft (or the application name given during PIA deployment)
4. Click on Web modules from Related Items table
5. Click on PORTAL module
6. Click on Session Management link from Additional Properties table
7. Check "Overwrite Session Management" flag
8. Click on “Enable Cookies” link
9. Enter a new cookie name for Cookie name field in the following format:

<machine-name>-<port>-PORTAL-PSJSESSIONID



User ID: lakshmi

LAGOURAB-PCNetwork

- ⊟ Servers
  - Application Servers
  - JMS Servers
  - Clusters
  - Cluster Topology
- ⊟ Applications
  - Enterprise Applications
  - Install New Application
- ⊞ Resources
- ⊞ Security
- ⊞ Environment
- ⊞ System Administration
- ⊞ Troubleshooting

The server may need to be restarted for these changes to take effect.

Enterprise Applications > peoplesoft > Web Module > PORTAL > Session Management

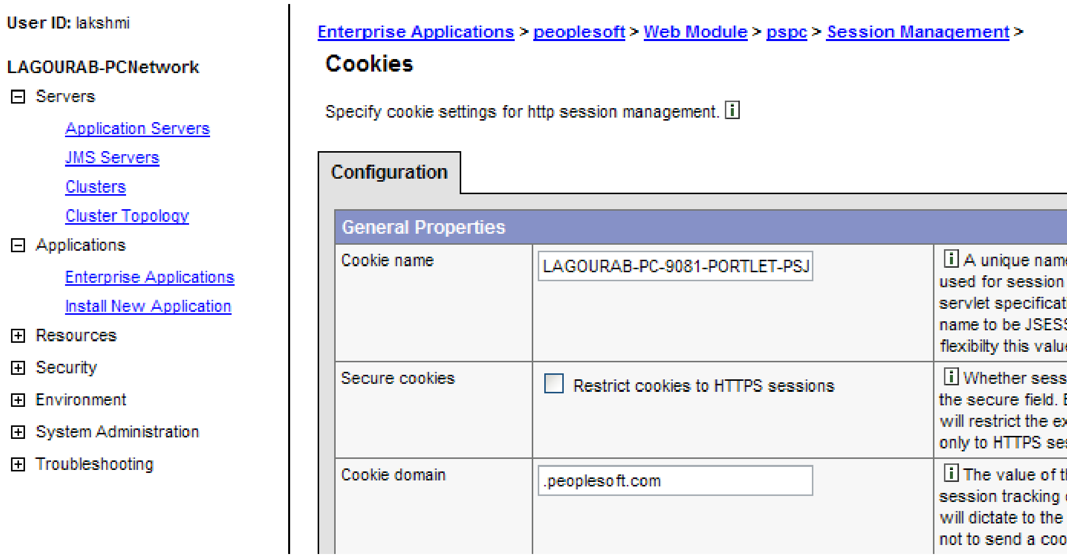
Cookies

Specify cookie settings for http session management. ℹ

Configuration

General Properties		
Cookie name	LAGOURAB-PC-9081-PORTAL-PSJS	ℹ A unique r used for sess servlet specif name to be JS flexibilty this v
Secure cookies	□ Restrict cookies to HTTPS sessions	ℹ Whether s the secure fie will restrict th only to HTTPS
Cookie domain	.peoplesoft.com	ℹ The value session track will dictate to

10. Enter the Cookie domain value if required
11. Click Apply and save the configuration
12. Repeat the same above steps for “pspc” module also. Change the cookie name to the following format : <machine>-<port>-PORTLET-PSJESSIONID



Cookies

Specify cookie settings for http session management.

General Properties		
Cookie name	LAGOURAB-PC-9081-PORTLET-PSJ	A unique name used for session servlet specificat name to be JSESS flexibilty this valu
Secure cookies	☐ Restrict cookies to HTTPS sessions	Whether sess the secure field. E will restrict the ex only to HTTPS ses
Cookie domain	.peoplesoft.com	The value of t session tracking will dictate to the not to send a coo

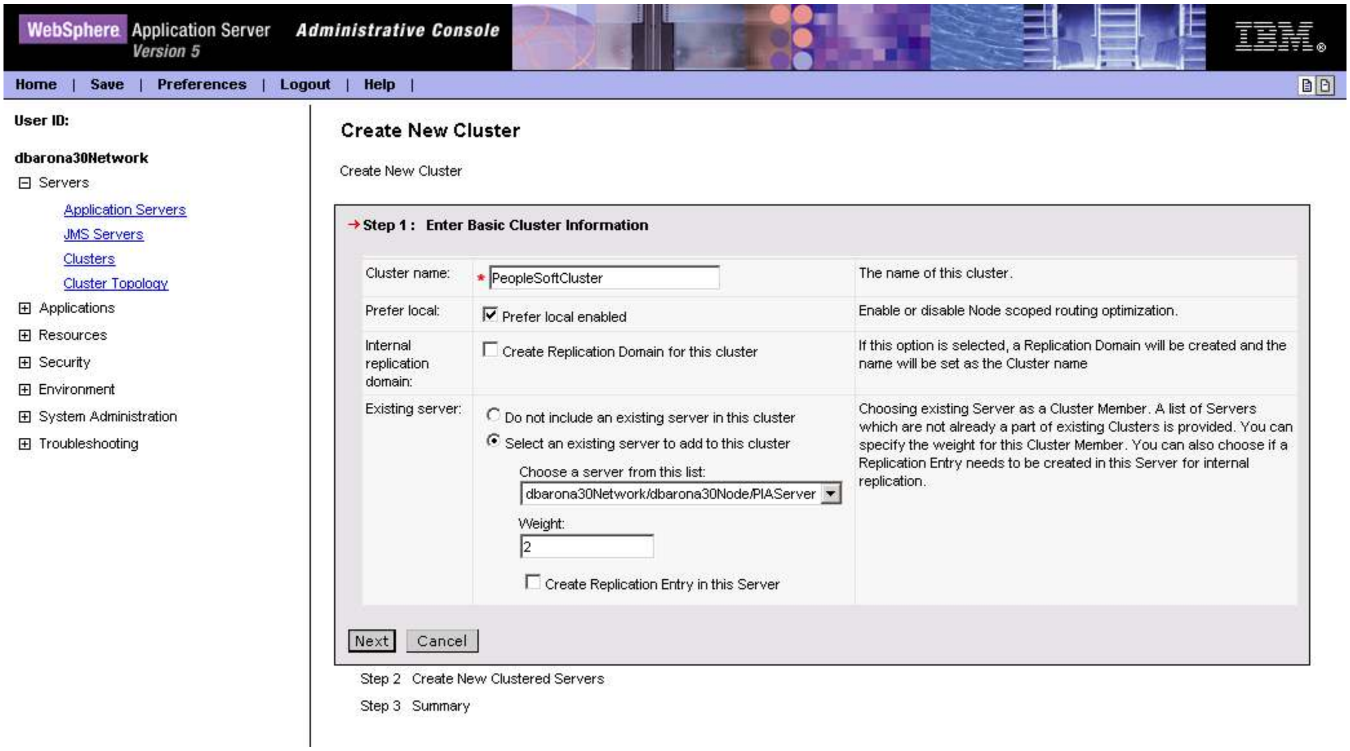
xii. The psreports directory is usually created as part of the PIA installation. Create the psreports directory on each node machine if it does not exist.

c. Select *Servers -> Clusters*

d. Click *New* to create a new cluster.

e. Call the new cluster *PeopleSoftCluster*. *Select an existing server to add to this cluster* and choose the *PIAServer* created in the previous step. This will apply the configuration changes to apply members in the cluster.

Note: Adding a server to a cluster is not reversible. A server cannot be removed from a cluster, only deleted from the cluster.



Create New Cluster

Create New Cluster

→ Step 1 : Enter Basic Cluster Information

Cluster name:	* PeopleSoftCluster	The name of this cluster.
Prefer local:	☑ Prefer local enabled	Enable or disable Node scoped routing optimization.
Internal replication domain:	□ Create Replication Domain for this cluster	If this option is selected, a Replication Domain will be created and the name will be set as the Cluster name
Existing server:	○ Do not include an existing server in this cluster ● Select an existing server to add to this cluster Choose a server from this list: dbarona30Network/dbarona30Node/PIAServer Weight: 2 □ Create Replication Entry in this Server	Choosing existing Server as a Cluster Member. A list of Servers which are not already a part of existing Clusters is provided. You can specify the weight for this Cluster Member. You can also choose if a Replication Entry needs to be created in this Server for internal replication.

Next Cancel

Step 2 Create New Clustered Servers

Step 3 Summary

Figure 3: Create New Cluster

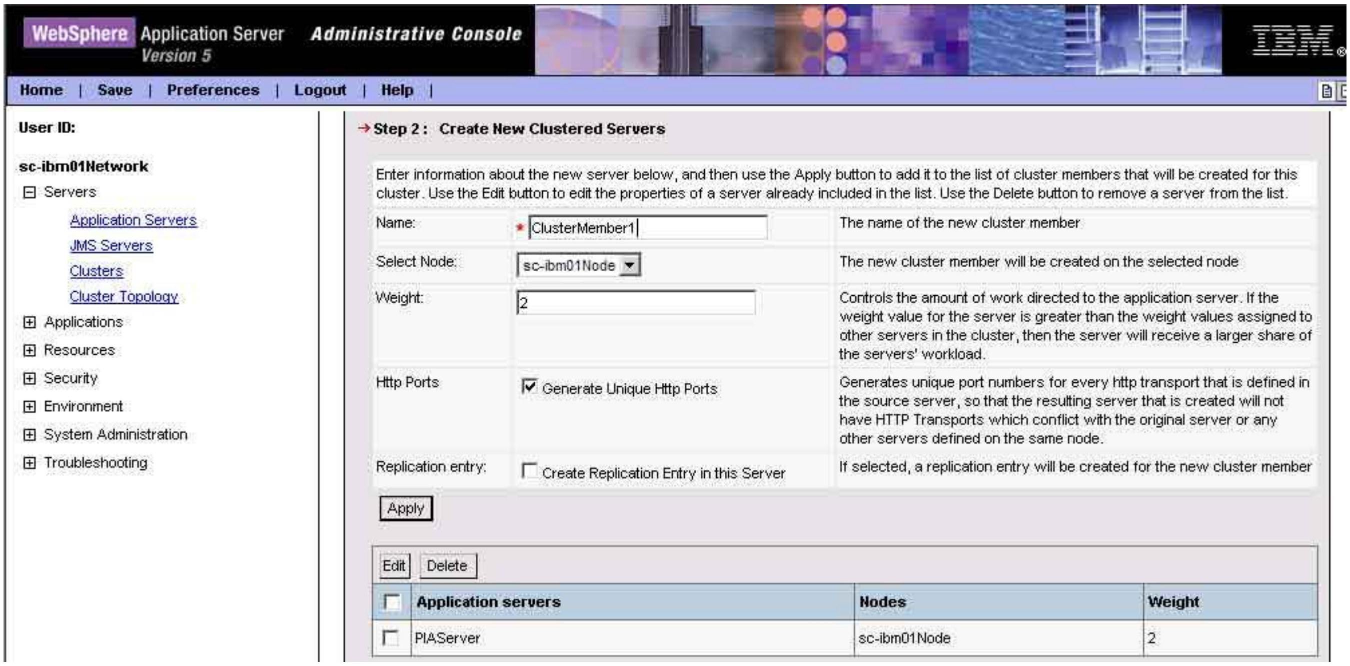
f. Click *Next*.

g. For each node to be added to the cluster,. Add one or more cluster members (depending on the number of application servers) to this cluster.

i. Enter a Name for the new cluster member (ex. ClusterMember1).

ii. From the drop down list, select the *Node* that the cluster member is to be associated with. Remember that a cluster can be deployed across multiple nodes with multiple cluster members on each node. If using different nodes, make sure to select a different node for the cluster members.

iii. Select the checkbox to *Generate Unique Http Ports*. WebSphere will assign unique ports to the new cluster member (these ports should be double checked to prevent interference with other applications' ports.



→ Step 2 : Create New Clustered Servers

Enter information about the new server below, and then use the Apply button to add it to the list of cluster members that will be created for this cluster. Use the Edit button to edit the properties of a server already included in the list. Use the Delete button to remove a server from the list.

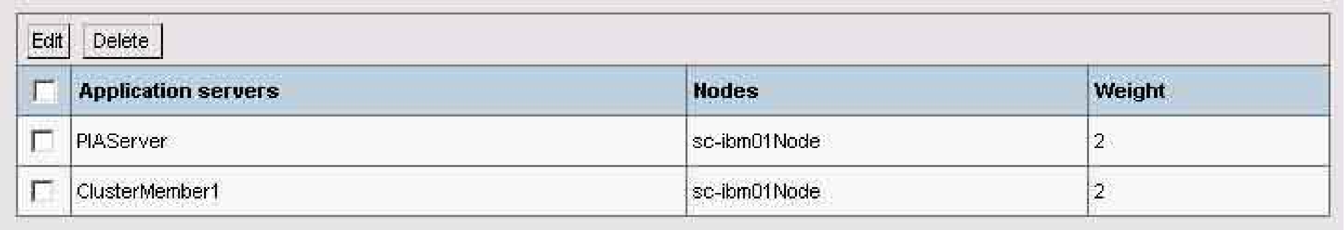
Name:	* ClusterMember1	The name of the new cluster member
Select Node:	sc-ibm01Node	The new cluster member will be created on the selected node
Weight:	2	Controls the amount of work directed to the application server. If the weight value for the server is greater than the weight values assigned to other servers in the cluster, then the server will receive a larger share of the servers' workload.
Http Ports	☑ Generate Unique Http Ports	Generates unique port numbers for every http transport that is defined in the source server, so that the resulting server that is created will not have HTTP Transports which conflict with the original server or any other servers defined on the same node.
Replication entry:	☐ Create Replication Entry in this Server	If selected, a replication entry will be created for the new cluster member

☐	Application servers	Nodes	Weight
☐	PIAServer	sc-ibm01Node	2

Figure 4: Create New Clustered Servers

iv. Click *Apply*.

v. The new cluster member will appear in the application server list.



☐	Application servers	Nodes	Weight
☐	PIAServer	sc-ibm01Node	2
☐	ClusterMember1	sc-ibm01Node	2

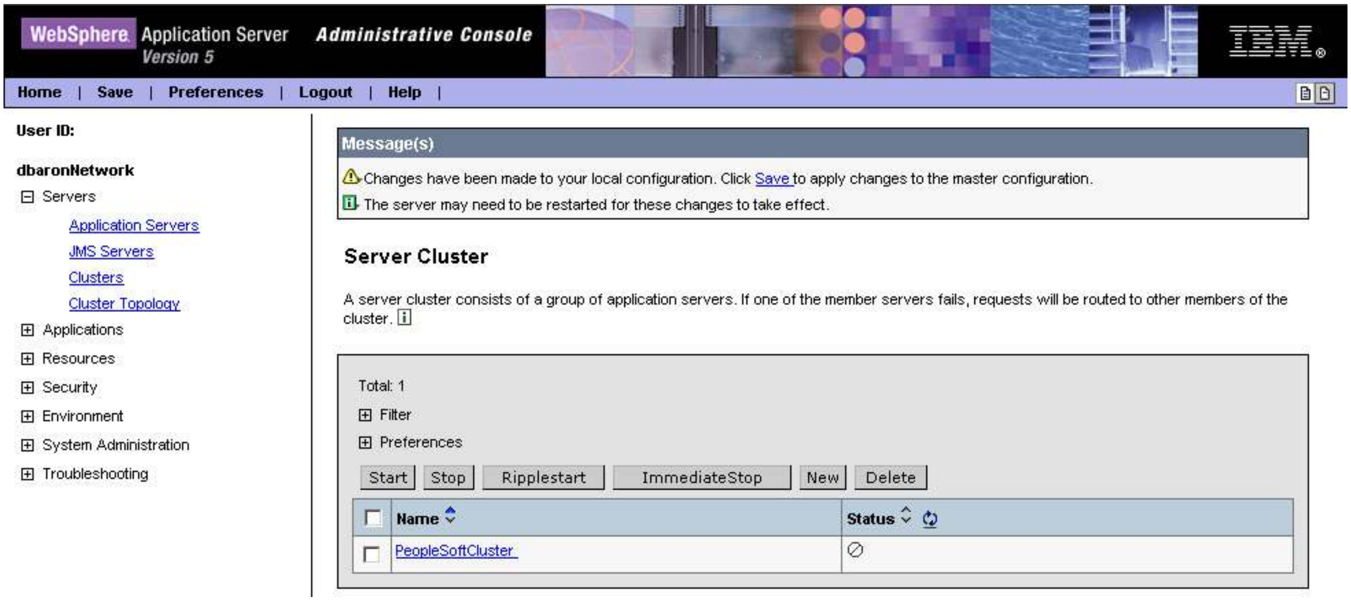
Figure 5: Application Server list

vi. Repeat steps i-v to add one or more cluster members on each node. Each cluster member will act as an application server as shown in figure 3 in RunTime Architecture. For best performance, balance the cluster members across multiple nodes.

vii. When finished adding cluster members, click *Next*.

viii. A summary of the changes will appear.

ix. Click *Finish* to create the cluster members.



Message(s)

⚠ Changes have been made to your local configuration. Click Save to apply changes to the master configuration.

The server may need to be restarted for these changes to take effect.

Server Cluster

A server cluster consists of a group of application servers. If one of the member servers fails, requests will be routed to other members of the cluster.

Total: 1

⊞ Filter

⊞ Preferences

□	Name	Status
□	PeopleSoftCluster	⊘

Figure 6: Save Cluster Member

x. You will see a warning message that changes have been made. You will need to save the changes by clicking the Save link and saving the configuration. Be sure to select *Synchronize changes with Nodes* to allow the nodes to be updated with changes.

h. Install the PeopleSoft PIA.

i. Select *Applications -> Install New Application*.

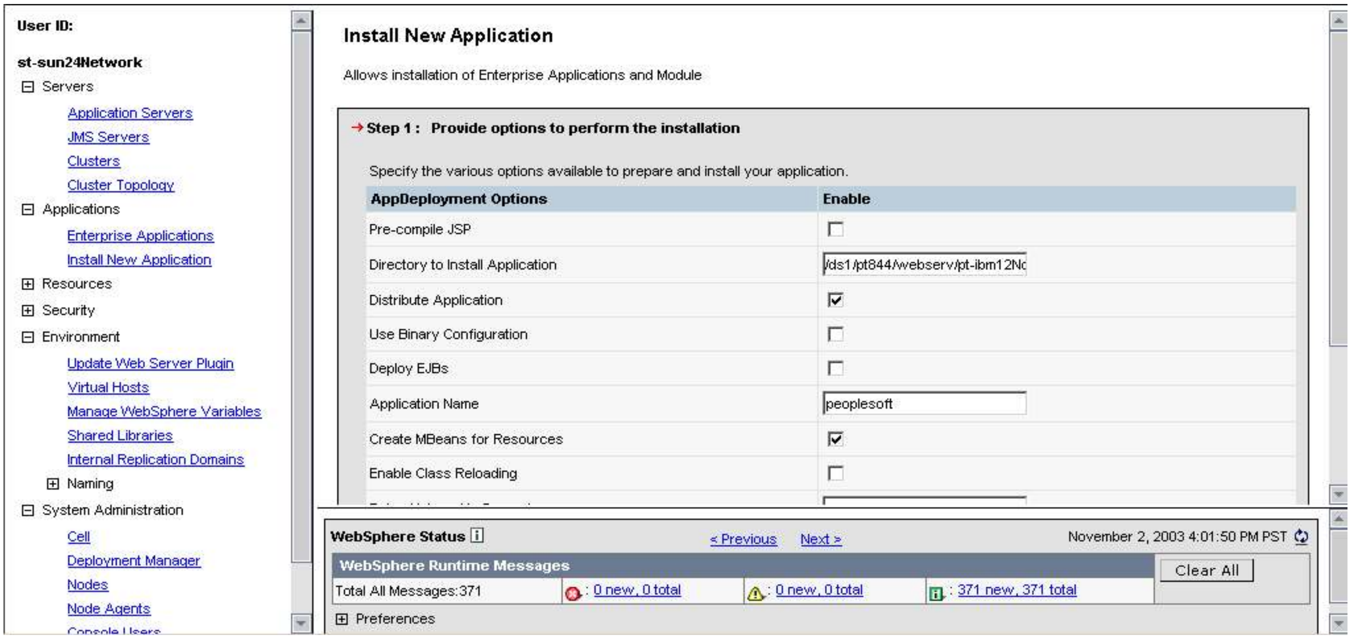
ii. Select Browse and navigate to the local path containing the ear file (the ear file was created in step 4). Select the ear file.
Windows
`C:\PS_HOME\PeopleSoft.ear`
Unix
/PS_HOME/PeopleSoft.ear

iii. Click *Next*. This step will take sometime as the ear file is uploaded to WebSphere.

iv. Click *Next* to advance to the Preparing for the application installation step.

v. Using the values recorded in step 4b during the PIA install, set “Directory to Install Application” to PS_HOME\webserv\<WebSphereNodeDirectory>

which was the directory that the PIA was installed to on the staging machine. This value must remain the same because the PIA contains hard coded references to directory structures. Change *Application Name* to the value specified in step 4b.
Note:If you want to enter any PS_HOME and application name, then follow either of two steps below
(a)Open web.xml file on respective base instances at PS_HOME/webserv/<cell_node_server>/<application_name>/PORTAL/web.xml and change <param-value> in <init-param> to appropriate PS_HOME.Similar change can be made to web.xml for PSIGW Web Application.
(b)Use Symbolic links to point to PS_HOME captured in step 4b.



Install New Application

Allows installation of Enterprise Applications and Module

→ **Step 1 : Provide options to perform the installation**

Specify the various options available to prepare and install your application.

AppDeployment Options	Enable
Pre-compile JSP	□
Directory to Install Application	/ds1/pt844/webserv/pt-ibm12Nc
Distribute Application	☑
Use Binary Configuration	□
Deploy EJBs	□
Application Name	peoplesoft
Create MBeans for Resources	☑
Enable Class Reloading	□

Figure 7: Install New Application

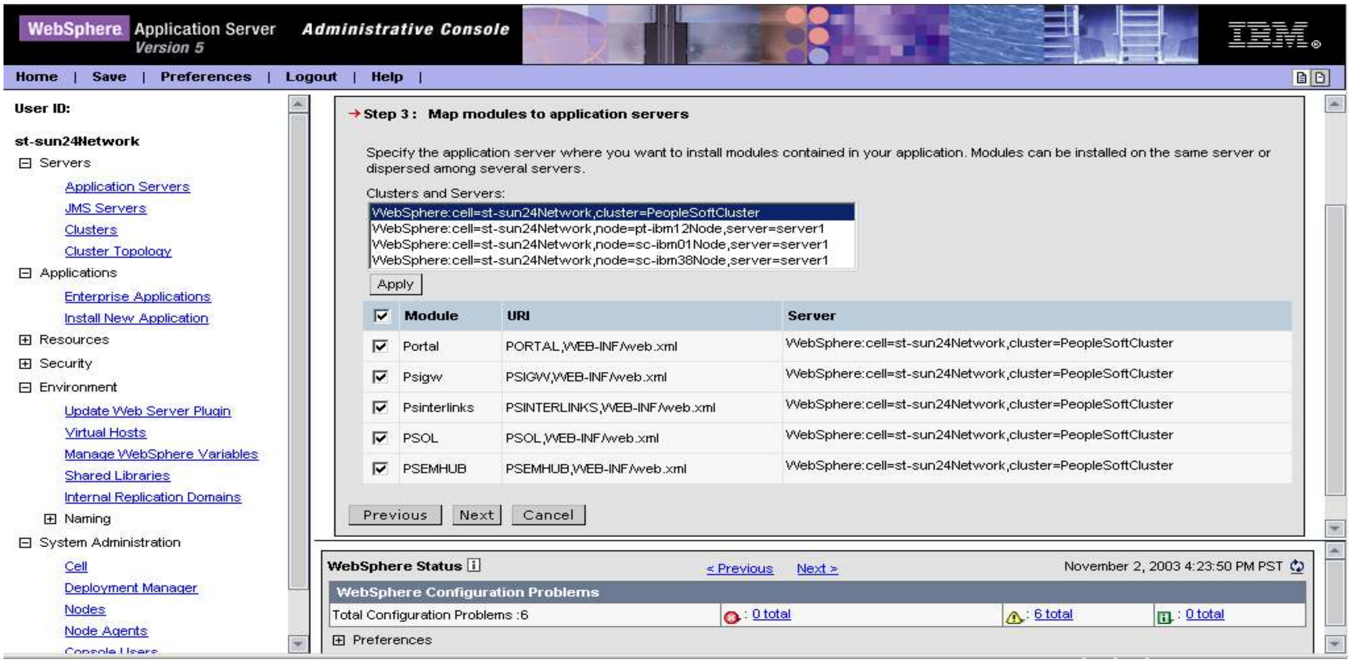
vi. Click *Next.*

vii. Accept the *default_host* as the *Virtual Host* for the Web Modules. No changes are required.

viii. Click *Next*.

ix. In "Map modules to application servers" check all the listed modules and select the *PeopleSoftCluster* in the Clusters and Servers listbox (see image below).

x. Click **Apply**. The server mappings should change to match the cluster name. To verify, look in the third column titled *Server*.



→ **Step 3 : Map modules to application servers**

Specify the application server where you want to install modules contained in your application. Modules can be installed on the same server or dispersed among several servers.

Clusters and Servers:

WebSphere:cell=st-sun24Network,cluster=PeopleSoftCluster
WebSphere:cell=st-sun24Network,node=pt-ibm12Node,server=server1
WebSphere:cell=st-sun24Network,node=sc-ibm01Node,server=server1
WebSphere:cell=st-sun24Network,node=sc-ibm38Node,server=server1

Apply

☑	Module	URI	Server
☑	Portal	PORTAL,WEB-INF/web.xml	WebSphere:cell=st-sun24Network,cluster=PeopleSoftCluster
☑	Psigw	PSIGW,WEB-INF/web.xml	WebSphere:cell=st-sun24Network,cluster=PeopleSoftCluster
☑	Psinterlinks	PSINTERLINKS,WEB-INF/web.xml	WebSphere:cell=st-sun24Network,cluster=PeopleSoftCluster
☑	PSOL	PSOL,WEB-INF/web.xml	WebSphere:cell=st-sun24Network,cluster=PeopleSoftCluster
☑	PSEMHUB	PSEMHUB,WEB-INF/web.xml	WebSphere:cell=st-sun24Network,cluster=PeopleSoftCluster

Previous Next Cancel

WebSphere Status < Previous Next > November 2, 2003 4:23:50 PM PST

WebSphere Configuration Problems

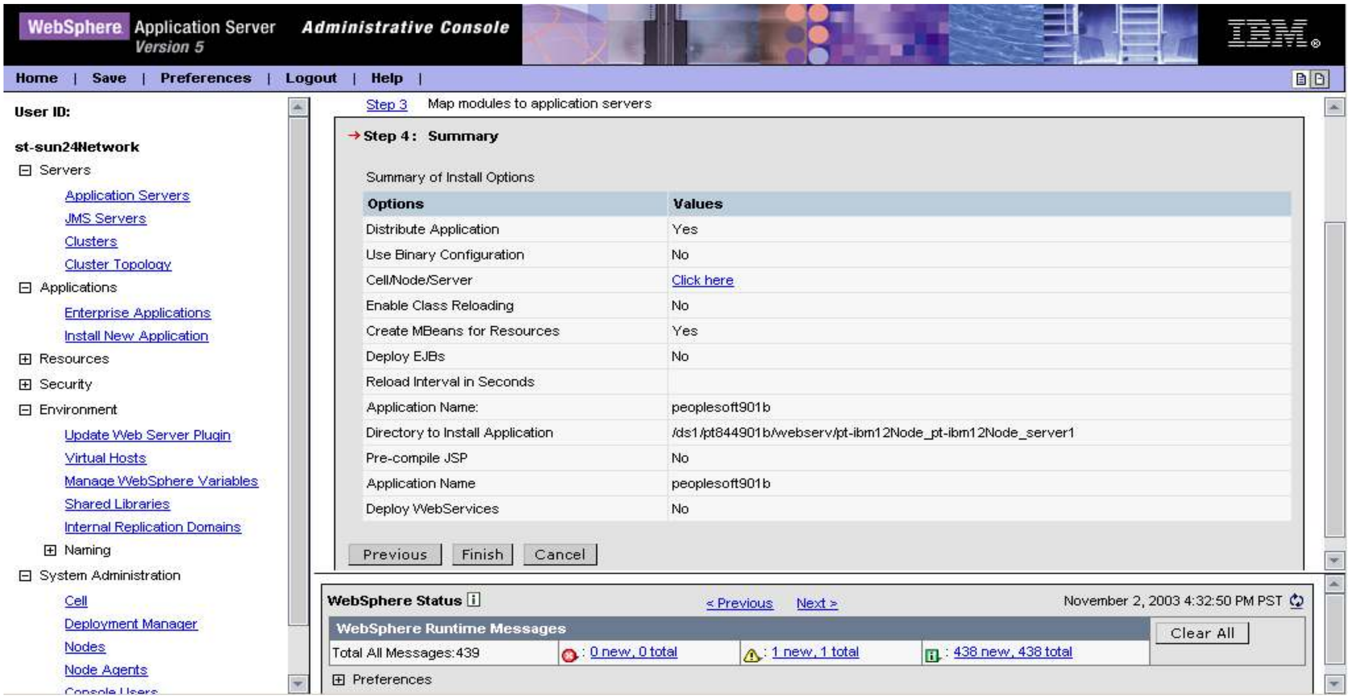
Total Configuration Problems :6 | ⊗ : 0 total | ⚠ : 6 total | ℹ : 0 total

⊞ Preferences

Figure 8: Map modules to application servers

xi. Click *Next.*

xii. On the Summary screen, review your choices. If a problem is seen, click *Previous* to correct the error. Make sure that the *application name* and the *Directory to Install Application* match the values used in the PIA installation in step 4. Click *Finish*. This step will take a few minutes to copy the ear file to the Deployment Manager.



Summary of Install Options

Options	Values
Distribute Application	Yes
Use Binary Configuration	No
Cell/Node/Server	Click here
Enable Class Reloading	No
Create MBeans for Resources	Yes
Deploy EJBs	No
Reload Interval in Seconds	
Application Name:	peoplesoft901b
Directory to Install Application	/ds1/pt844901b/webserv/pt-ibm12Node_pt-ibm12Node_server1
Pre-compile JSP	No
Application Name	peoplesoft901b
Deploy WebServices	No

WebSphere Status < Previous Next > November 2, 2003 4:32:50 PM PST

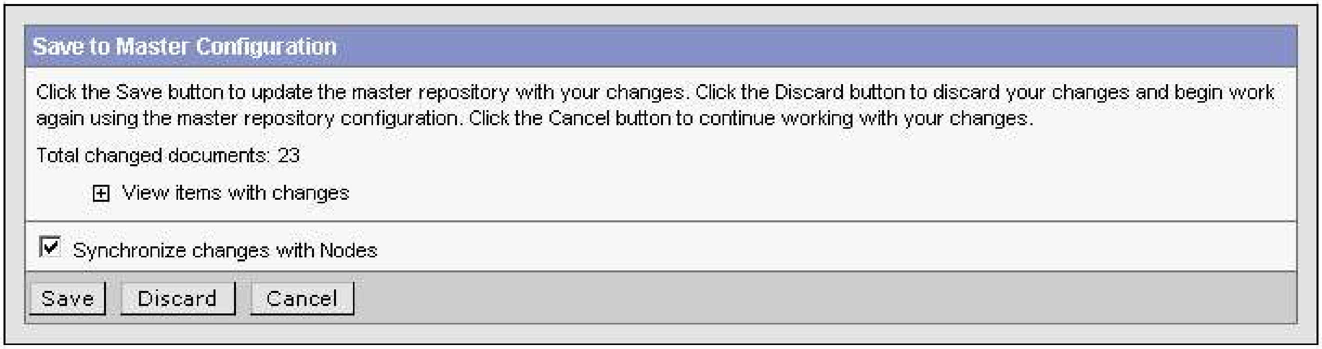
WebSphere Runtime Messages

Total All Messages:439 | : 0 new, 0 total | : 1 new, 1 total | : 438 new, 438 total

Figure 9: Summary

xiii. Select *Save to Master Configuration*. You will see a page that displays the status of the PeopleSoft application install as successful.

xiv. Check the box “Synchronize changes with Nodes”



Click the Save button to update the master repository with your changes. Click the Discard button to discard your changes and begin work again using the master repository configuration. Click the Cancel button to continue working with your changes.

Total changed documents: 23

☑ Synchronize changes with Nodes

Figure 10:Save to Master Configuration

xv. Click “Save” button. This step will take a few minutes to copy the application to all the nodes.

xvi. The Configuration for all servers will be updated, and the EAR file will be transferred and unpacked on all attached servers.

xvii. Logout from the Administrative console.

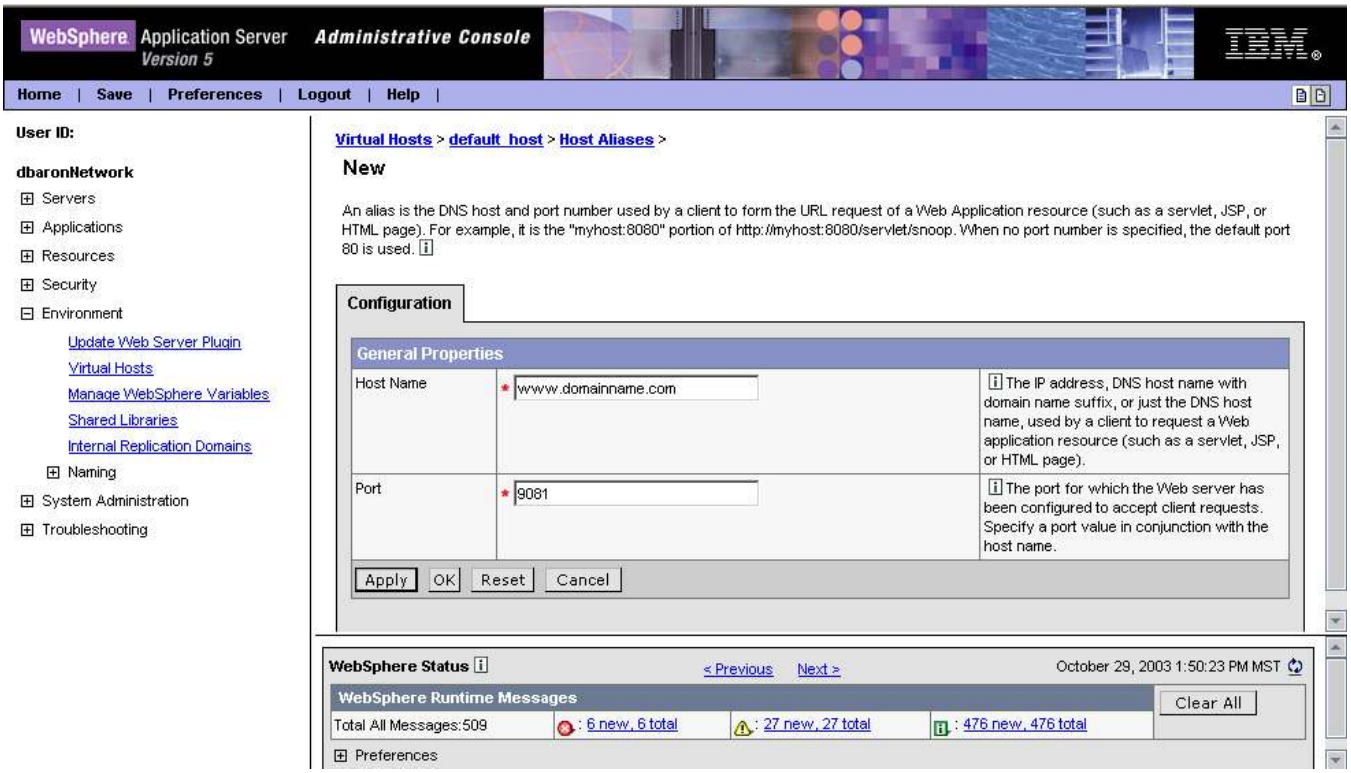
i. If not using an HTTP Server (like IHS, Sun One, or IIS), verify that any HTTP ports that the servers are using have been added the Virtual Hosts for the default_host (or the new virtual host created for the PIA). If using an HTTP Server, skip to step j.

i. Select *Environment -> Virtual Hosts*.

ii. Select *default_host*.

iii. Select *Host_Aliases*. You will see the list of the hosts and ports defined for the *default_host*. The Host Name values should be changed to specify hostnames instead of wildcards (*). The HTTP port for each cluster member should also be added (the default is to increment by 1 starting at 9080).

iv. To add a new host/port combination, click the *New* button.



Virtual Hosts > default_host > Host Aliases >

New

An alias is the DNS host and port number used by a client to form the URL request of a Web Application resource (such as a servlet, JSP, or HTML page). For example, it is the "myhost:8080" portion of http://myhost:8080/servlet/snoop. When no port number is specified, the default port 80 is used.

Configuration

General Properties		
Host Name	* www.domainname.com	The IP address, DNS host name with domain name suffix, or just the DNS host name, used by a client to request a Web application resource (such as a servlet, JSP, or HTML page).
Port	* 9081	The port for which the Web server has been configured to accept client requests. Specify a port value in conjunction with the host name.

October 29, 2003 1:50:23 PM MST

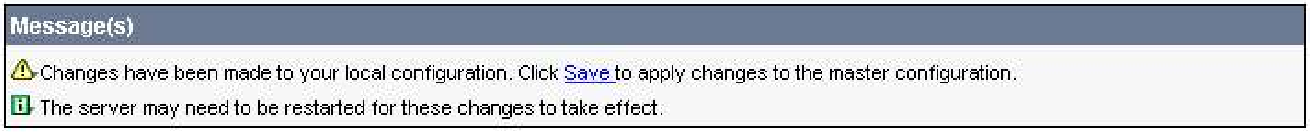
WebSphere Runtime Messages			
Total All Messages:509	: 6 new, 6 total	: 27 new, 27 total	: 476 new, 476 total

Figure 11:Virtual Host

v. Enter an appropriate *Host Name* and *Port* in the text boxes. Click OK when finished.

vi. The host and port will be added the list of *Host Aliases*. You will see a warning message that changes have been made. You will need to save the changes by clicking the Save link and saving the configuration. The servers will need to be restarted to recognize the new host(s) and port(s).

vii. Skip to step k if not using an HTTP server.



Message(s)

⚠ Changes have been made to your local configuration. Click Save to apply changes to the master configuration.

ℹ The server may need to be restarted for these changes to take effect.

Figure 12:Save Message

j. Update the HTTP server plugin

i. Select Environment->Update Web Server Plugin

ii. Click "OK"

iii. To view the plugin configuration, click on "View or download the current web server plugin configuration file"

iv. The plugin-cfg.xml file must be copied to the HTTP Server machine and placed in the correct directory. To determine the directory, refer to the previous section of this document titled "Configuring a WebSphere HTTP (proxy) Server with PeopleTools 8.44". To obtain the plugin-cfg.xml file, right click on "View or download the current web server plugin configuration file" and "Save Target As…". Copy plugin-cfg.xml to the directory found in the previous step.

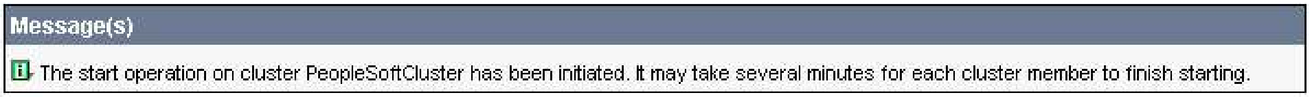
v. Your PeopleSoft applications will now get load balancing and failover when accessed through the HTTP server.

k. Start the Cluster

   i. Select *Servers->Clusters*.

   ii. Check PeopleSoftCluster.

   iii. Click "Start".



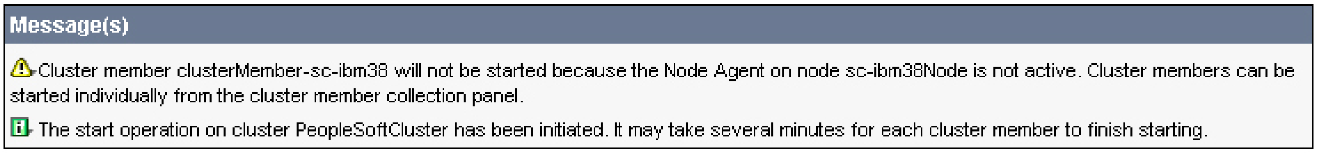
Message(s)

The start operation on cluster PeopleSoftCluster has been initiated. It may take several minutes for each cluster member to finish starting.

Figure 13:Cluster message

   iv. When all of the servers in the cluster have started the status icon will update to a solid green arrow.

   v. If a message appears warning that a cluster member will not be started, check that the node agent is running on that machine (use the startNode.[bat|sh] command located in <WAS_HOME>/bin). Once that node agent is started, select the cluster and click start to start the cluster member.



Message(s)

⚠ Cluster member clusterMember-sc-ibm38 will not be started because the Node Agent on node sc-ibm38Node is not active. Cluster members can be started individually from the cluster member collection panel.

The start operation on cluster PeopleSoftCluster has been initiated. It may take several minutes for each cluster member to finish starting.

Figure 14: Cluster Message

   vi. Logout from the Administrative console.

Cluster Topology

WebSphere provides a topology overview of the cluster. All the servers in the cluster can be viewed from within the DeploymentManager. The nodes and their cluster members are shown.

1. From any machine, start the WebSphere Administrative Console *pointing to the DeploymentManager.*

2. Select *Servers->Cluster Topology*.

3. Expand the *Cell* in the right pane.

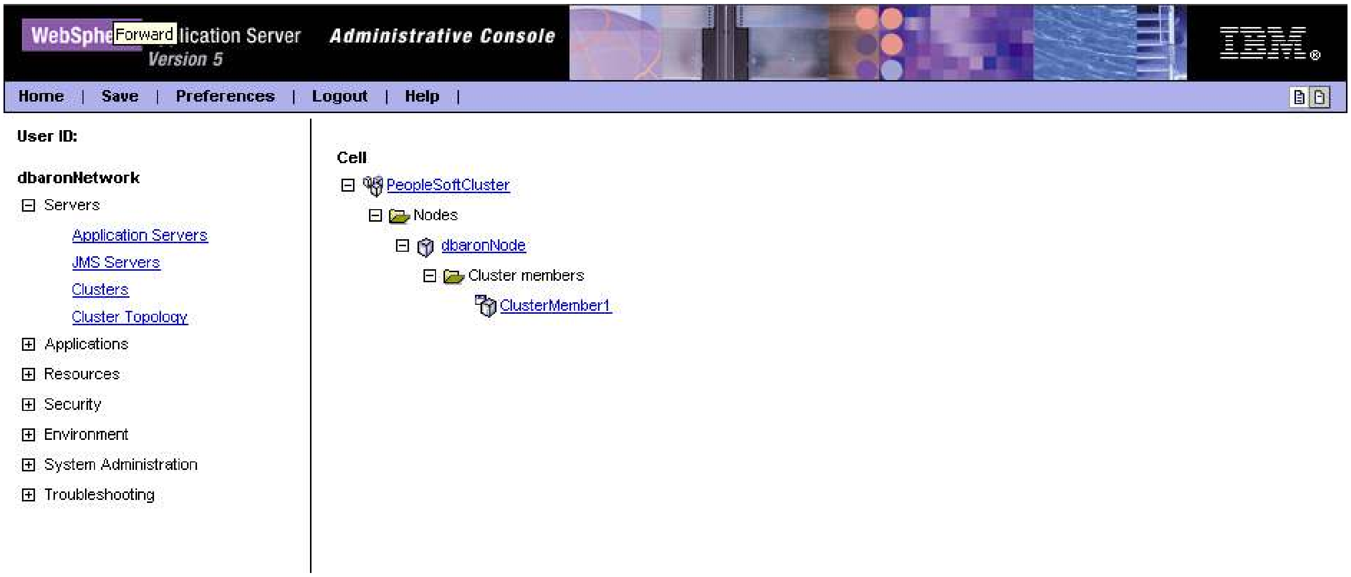


Figure 15: Cluster Topology

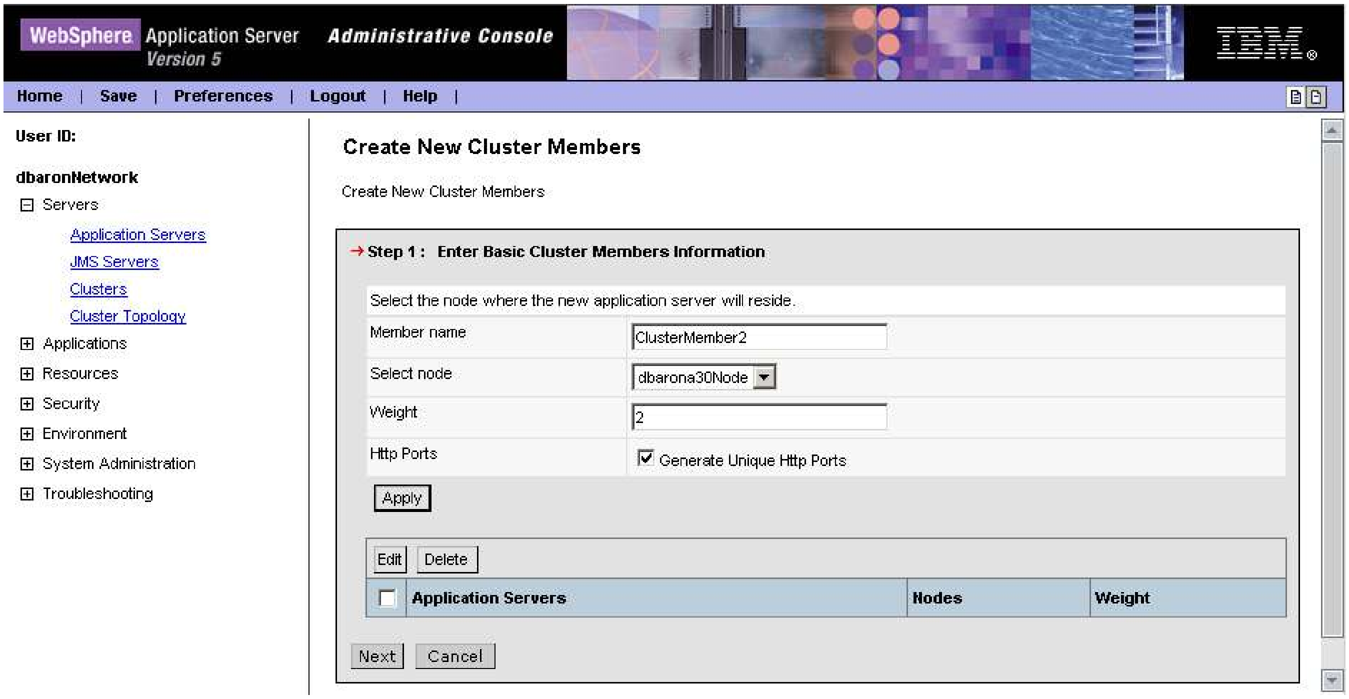
4. The cluster topology shows the nodes and cluster members configured for a cluster.

Adding an Additional Cluster Member

After configuring a cluster, additional cluster members may be added to the cluster to provide more resources to handle client requests.

1. From any machine, start the WebSphere Administrative Console *pointing to the DeploymentManager.*
2. Select *Servers->Clusters*.
3. Select *PeopleSoftCluster*.
4. Select *Cluster Members*.

5. Select *New*.



Create New Cluster Members

Create New Cluster Members

→ Step 1 : Enter Basic Cluster Members Information

Select the node where the new application server will reside.

Member name	ClusterMember2
Select node	dbarona30Node
Weight	2
Http Ports	☑ Generate Unique Http Ports

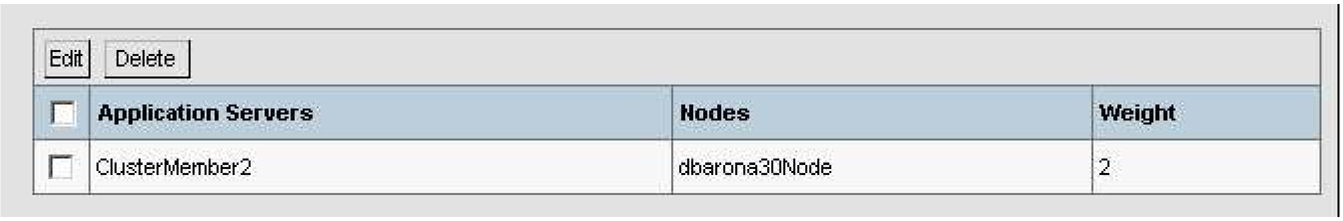
□	Application Servers	Nodes	Weight

Figure 16: Create New Cluster Member

6. Enter a *Member name* for the new cluster member.

7. Select the node where to create the new cluster member from the drop down list.

8. Click *Apply* to add the cluster member to the *Application Servers* list.



□	Application Servers	Nodes	Weight
□	ClusterMember2	dbarona30Node	2

Figure 17: Cluster member list

9. Repeat steps 7-9 to add as many cluster members as desired.

10. Click *Next* to advance to the summary screen.

11. Review the changes, and select *Finish*.

12. The changes must be saved. Click Save and then save the changes. Be sure to select the checkbox next to *Synchronize changes with Nodes*. Synchronizing the changes will copy the application's files to the node (ex. C:\PS_HOME will be created).

13. Expand *Servers -> Cluster Topology*.
14. The node agents must be running to start the cluster members on a node.
15. Start the cluster from *Servers -> Clusters*.

Updating PIA with a PeopleTools patch in clustered environment

Remember that the PIA can only be installed on a WebSphere Base instance. It is recommended to have a separate WebSphere Base instance available to install the new PIA with the patches. This instance is used as the staging server that will allow updates to be applied to the PIA PeopleTools patches which can be applied to the staging server (ex. Upgrading from PeopleTools 8.44.00 to 8.44.05**)**. **Note: For patch updates, you must use the same staging server that you used to originally to create the existing WebSphere installed PeopleTools application**. **This is due to the specific hardcoded node/server name reference in the PS_HOME path by the PIA installer that is required.**

An updated ear file can be exported from the staging server. This ear file can be copied to the production (or test environment). This new ear file can be used to update the existing cluster with the latest patched PIA version. Follow these steps to update the cluster with the patched PIA.

1. On the staging server with the WebSphere Base instance, apply the PeopleSoft Patches. Follow PeopleSoft provided instructions for installing the Patch.
2. Install the PIA using the mpinternet installer from Peoplesoft to the staging WebSphere instance. **Enter the PS_HOME and application name of the PIA application already deployed.** Retrieve these values from the web.xml file in the deployed cluster.
    a. Open PSHOME/webserv/<WebSphereNodeDirectory>/webserv/peoplesoft.ear/PORTAL/WEB-INF/web.xml.
    b. Record PSHOME and the application name as shown in this example: <servlet-name>psc</servlet-name>

```
<servlet-class>psft.pt8.psc</servlet-class>
<init-param id="InitParam_1078439625518">
  <param-name>configDir</param-name>
  <param-value>
```

C:\PS_HOME\webserv\dbarona30node_dbarona30node_ser

ver1/**peoplesoft**.ear/PORTAL/WEB-INF/psftdocs</param-value>

In the above example, PS_HOME is **C:\PS_HOME\webserv\dbarona30node_dbarona30node_server1** & application name is **peoplesoft** without 'ear' extension.

3. Test the changes to make sure the PIA is working as expected.

4. Use the EARExpander to create the ear file.

   b. Execute WebSphere's EARExpander from WAS_HOME/bin to create an EAR file with the customizations. The operationDir value should be /PS_HOME/webserv/<WebSphereNodeDirectory>/<application name>. If we consider above example, PS_HOME will be **C:\PS_HOME\webserv\dbarona30node_dbarona30node_server1.**

      These commands should be typed on a single line.

      Windows

      ```
      C:\<WAS_HOME>\bin\EARExpander.bat -ear
      C:\PS_HOME\PeopleSoft.ear -operationDir
      C:\PS_HOME\webserv\dbarona30node_dbarona30node_
      server1\peoplesoft.ear -operation collapse
      ```

      Unix

      ```
      /<WAS_HOME>/bin\EARExpander.sh -ear
      /PS_HOME/PeopleSoft.ear -operationDir
      /PS_HOME/webserv/
      dbarona30node_dbarona30node_server1/DeployedPeo
      pleSoft.ear -operation collapse
      ```

5. Copy the ear file from the staging server to a production server.

6. Open a browser and access the WebSphere Administrative Console *pointing to the DeploymentManager* e.g. http://<DeploymentManager>:9090/admin.

   a. Select *Applications -> Enterprise Applications*.

b. Select the *peoplesoft* application checkbox, and click *Update*.

c. Click *Browse* and navigate to the location of the ear file.

d. Click *Next* to continue the installation.

e. Click *Next* to continue the installation.

f. Specify PSHOME used in the initial installation from step 2 of this section.

   (Refer **Figure # 7**:Note: You can't update PeopleSoft application since you select it to update)

g. Click *Next*.

h. Accept the default value, *default_host*. Click *Next*.

i. Verify that the PIA is set to install to the cluster. Click *Next*.

   (Refer **Figure # 8**)

j. Click *Finish* to complete the update of the PIA.

k. *Save* the changes to the *Master configuration* and synchronize the changes to the nodes.

   (Refer **Figure # 10**)

l. The updated PIA has been installed to the cluster.

m. The cluster is ready to serve requests.

Updating PIA with a POC in clustered environment

The PIA can only be installed on a WebSphere Base instance. It is recommended to have a separate WebSphere Base instance available to install the PIA. This instance can be used as a staging server that will allow updates to be applied to the PIA. POCs can be applied to the staging server. An updated ear file can be exported from the staging server. This ear file can be copied to the production (or test environment) and used to update the application there. The following steps show the process.

1. You will need the values for PSHOME and the application name specified during the PIA installation. **Enter the PS_HOME and application name of the PIA application already deployed.** Retrieve these values from the web.xml file in the deployed cluster.

   a. Open PS_HOME/webserv/<WebSphereNodeDirectory>/webserv/peoplesoft.ear/PORTAL/WEB-INF/web.xml.

b. Record PSHOME and the application name as shown in this example: <servlet-name>psc</servlet-name>

<servlet-class>psft.pt8.psc</servlet-class>

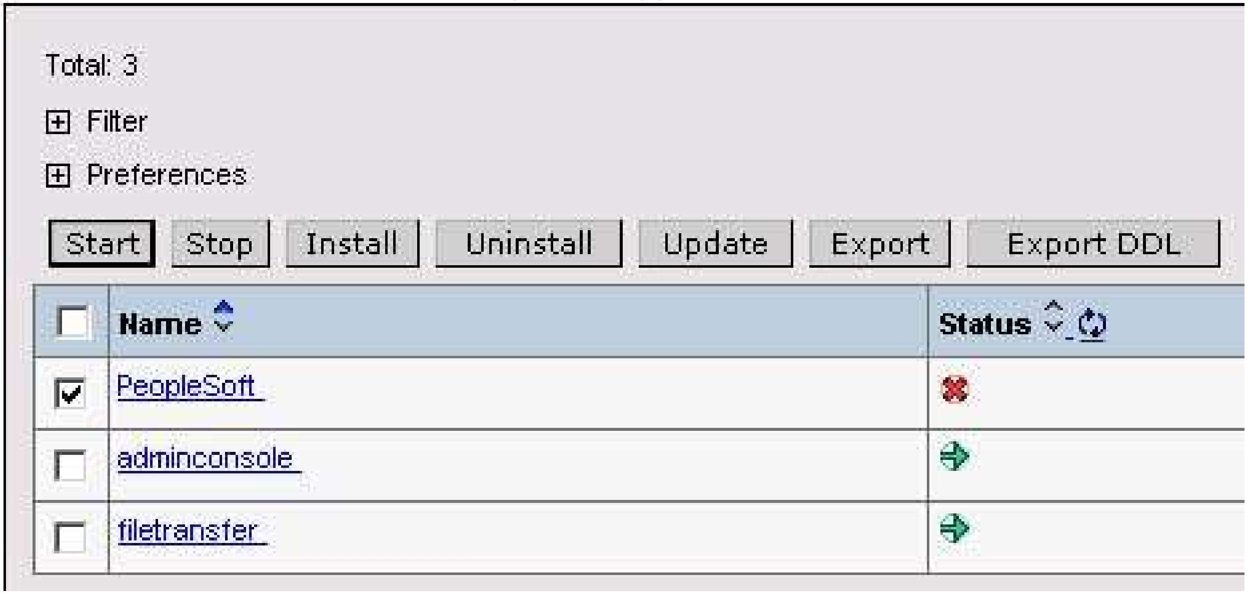
<init-param id="InitParam_1078439625518">

<param-name>configDir</param-name>

<param-value> **C:\PS_HOME\webserv\dbarona30node_dbarona30node_server1**/**peoplesoft**.ear/PORTAL/WEB-INF/psftdocs</param-value>

In the above example, PS_HOME is **C:\PS_HOME\webserv\dbarona30node_dbarona30node_server1** & application name is **peoplesoft** without ‘ear’ extension.

2. Stop the cluster from the WebSphere Admin Console.
3. In the Admin Console, open *Applications* -> *Enterprise Applications*.
4. Select the checkbox next to the PeopleSoft application.

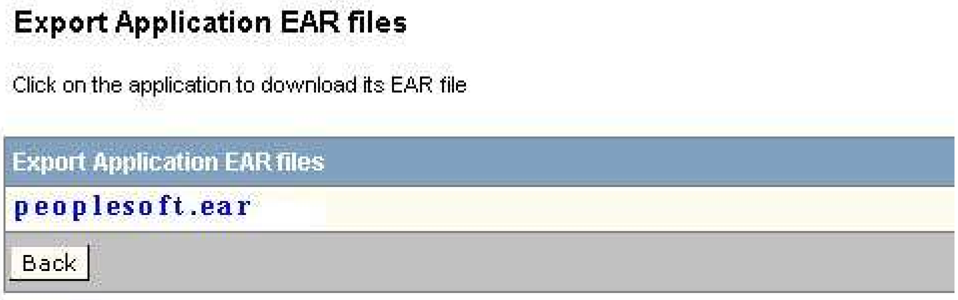


□	Name	Status
☑	PeopleSoft	✖
□	adminconsole	➔
□	filetransfer	➔

Figure 18: Enterprise Application

5. Click *Export.* WebSphere will build the ear file.

6. Click on the ear file link and save the file to a local disk (ex. C:\temp)



Export Application EAR files

Click on the application to download its EAR file

Export Application EAR files
peoplesoft.ear

Figure 19:Export PeopleSoft application

7. Run EARExpander from WAS_HOME/bin to expand the EAR file to a local directory (ex. c:\temp\POCEARExpand)

   Sample command
   C:\WebSphere51\AppServer\bin>EARExpander -ear peoplesoft.ear -operationDir c:\temp\POCEARExpand -operation expand

8. Apply the POC to the expanded directory. Refer to POC instructions to determine the directory to copy the POC files into.

9. Collapse the peoplesoft application with POC into an ear using EARExpander tool.

   Sample command
   c:\temp>C:\WebSphere51\AppServer\bin\EARExpander -ear peoplesoftpoc<date>.ear -operationDir C:\temp\POCEARExpand -operation collapse

10. Copy the ear file from the staging server to a production server.

11. Open a browser and access the WebSphere Administrative Console *pointing to the DeploymentManager* e.g. http://<DeploymentManager>:9090/admin.

    c. Select *Applications -> Enterprise Applications*.

d. Make sure that the *Status* of the PIA is stopped. If the status is started, select the checkbox next to *peoplesoft* and click *Stop*. (Refer **Figure 18**)

e. Select the *peoplesoft* application checkbox, and click *Update*.

f. Click *Browse* and navigate to the location specified in step 1 above.

g. Click *Next* to continue the installation.

h. Click *Continue*. (If you don't get the Application Security Warnings screen, you can skip this step)

i. Specify the PSHOME directory used in the initial installation. (See step 1 for the PSHOME directory).

j. Click *Next*.

k. Accept the default value, *default_host*.

l. Click *Next*.

m. Verify that the PIA is set to install to the cluster. Click *Next*.

   (Refer **Figure # 8**)

n. Click *Finish* to complete the update of the PIA.

o. *Save* the changes to the *Master configuration* and synchronize the changes to the nodes.
   (Refer **Figure # 10**)

p. The updated PIA has been installed to the cluster.

12. Start the cluster to start the clustermembers with the updated PIA.

Managing the Cluster

Once the PIA cluster is configured, WebSphere can be used to manage the cluster. Please refer to the WebSphere InfoCenter and other WebSphere references for more detailed instructions.

CONFIGURING A WEBSPHERE CLUSTER WITH PEOPLETOOLS 8.49

Starting from PT 8.49, WebSphere ND 6.1.0.3 is supported. The configuration steps for setting up cluster are changed when WAS ND 6.1 is used. There are no separate installs of Base and ND required and you need to install WAS ND 6.1 and this contains the capability to create different profile types like *Deployment Manager*, *Application Server* and *Custom Profiles*. For more information, refer to *Working with WebSphere* section of *PeopleTools : Server Administration Guide*.

IBM WebSphere Application Server Network Deployment (ND) V6.1 – provides central administration of multiple WebSphere Application Server Base processes on one or more machines. This component contains mechanism to create different profile types which are different WebSphere JVM instances and we can create the profiles based on our requirement.

WebSphere 6.1 Clustering Process Overview

Starting from PT 8.49, WebSphere Application Server ND 6.1 is supported and there are no separate installs of Base and ND are required. In PT 8.49, PIA gets deployed to a single application server profile which gets created during the PIA install. To create a WebSphere cluster, a deployment manager profile needs to be created and we add the multiple nodes (which contains the servers) to the deployment manger.

Here is a high level overview of the clustering process. The step-by-step process follows in the next section. Assume you have machines *WebSphere 1 to WebSphere N* for cluster setup.

- Install WebSphere ND software on each of the machines *WebSphere 1 to WebSphere N*.
- Install the PIA on a single machine (e.g. *WebSphere 1*) and select single server installation. This will create an application server profile and deploys the PeopleSoft application on to the server.
- Create a deployment manager profile (cell) on a machine (Let's say *WebSphere1*)
- Add the node from *WebSphere 1* instance to DM
- On *WebSphere 2* to *WebSphere N instances*, Install Customer profiles(nodes) and add these nodes to the DM
- Create a cluster.
- Create cluster members using the nodes added in the DM.
- Deploy PeopleSoft Application.
- Target the PeopleSoft application to the cluster that you have created in previous steps.
- Make any necessary changes to the virtual hosts and generate the plugin-cfg.xml.
- Start the cluster.

> **Note:** On *WebSphere 1 to WebSphere N*, you can choose any machine to install DM.

Configuring a WebSphere Cluster using WAS ND 6.1

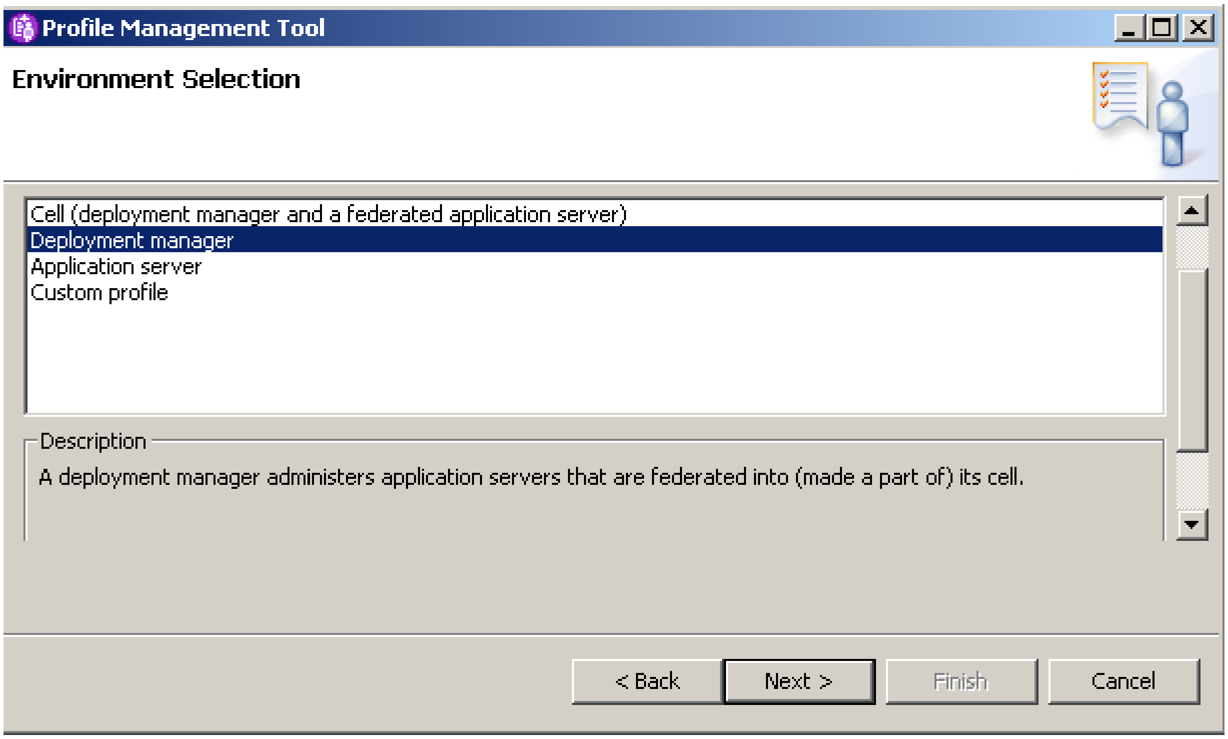
1. Install WebSphere Application Server ND 6.1 on multiple servers(*webSphere 1* to *WebSphere N*) as per your requirement. Use the instructions provided in the *PeopleTools 8.49 Install Guide*.

2. Install PIA from PT 8.49 and select *Single Server Installation* on one of the WebSphere servers (*WebSphere1).* Use the instructions provided in the *PeopleTools 8.49 Install Guide*

3. Select one of the server to host the *Deployment Manager* from the available WebSphere servers (*WebSphere1, WebSphere 2…..WebSphere N*).

4. On the machine which will host DM (Deployment Manager), assume that it is server *WebSphere1*, use *Profile Management Tool* and create a DM profile. Follow the steps listed below to create the DM profile,

   - Go to *WAS_HOME\bin\ProfileManagement\bin\* and run *PMT.bat (or .sh)* and this will bring up a profile creation tool wizard.
   - Select *Deployment manager* profile type and click *Next*



Cell (deployment manager and a federated application server)
Deployment manager
Application server
Custom profile

Description
A deployment manager administers application servers that are federated into (made a part of) its cell.

   - Select *Advanced Profile creation* option and click *Next*
   - Select *Deploy the administrative console* option and click *Next*

- Select a name for profile and the location where the profile will be created and click *Next*
- Keep the node name, cell name and hostname same and click *Next*
- Uncheck the *Enable administrative security* option and click *Next*
- Keep the port values as assigned and click *Next*
- Uncheck the *Run Deployment Manager as windows service* option and click *Next*
- Verify all the items on the *Profile Creation summary* page and click *create* button

This will create a *Deployment manager* profile and you can use it setup the cluster.

5. If using a webserver (like *IBM HTTP Server, Sun Java Webserver or IIS*) as *Reverse Proxy Server* in front of DM, install a supported webserver on *HTTPServer* machine. Use the instructions provided in the *PeopleTools 8.49 Install guide*.

6. On *WebSphere1*, install the PIA as mentioned in step 2, record settings.

a. Install PIA using the instructions in the *PeopleTools 8.49 Install guide*.

b. Record *PS_HOME, AuthTokenDomain, Application Name* and *psreports* path. This information is very important and should be saved for future reference.

Note: The value for *PS_HOME* will be used by all machines in the cluster, so make sure the path is valid for each machine. The same values of *PS_HOME* and application name need to be used when custom profiles are created on *WebSphere 2* to *WebSphere N* instances

c. Create any additional PeopleSoft sites.

d. Perform any required customizations.

e. Test to ensure setup works.

f. Execute WebSphere's *EARExpander* from *PS_HOME/webserv/<appname>/bin* to create an EAR file with the customizations.

```
Windows:
```

```
C:\<PS_HOME>\webserv\<appname>\bin\EARExpander.
bat -ear C:\<PS_HOME>\peoplesoft.ear -
operationDir
C:\<PS_HOME>\webserv\<appname>\installedApps\<h
ostname>NodeCell\<appname>.ear -operation
collapse

Unix:
/<PS_HOME>/
webserv/<appname>/bin/EARExpander.sh -ear
/<PS_HOME>/peoplesoft.ear -operationDir
/<PS_HOME>/webserv/<appname>/installedApps/<hos
tname>NodeCell/<appname>.ear -operation
collapse
```

Be sure the Deployment Manager is started on the machine where you installed DM profile

```
Windows: DeploymentManager_Home\bin\startManager.bat
```

```
Unix: DeploymentManager_Home/bin/startManager.sh
```

Where *DeploymentManager_Home* is the location of the DM profile that you have created on step 4.

7. On *WebSphere2* to *WebSphereN,* create *Custom* profiles and add these nodes to DM

a. Before federating the node, check that the date and times for the machines are within 5 minutes of each other. If the machines do not have similar times, an error message will appear and the node will not be added.

b. The *Deployment Manager* machine and the *Node Agent* machine must be able to resolve the hostnames of each other. To test, use the hostname with the ping command.

c. Create *Custom profiles* on machines *WebSphere 2* to *WebSphere N* using the *Profile Management Tool* as described in step 4, but select *Custom Profile* under *Environment Selection*. Enter the profilename same as the application name you have given during step 6 and set the profile path same as the PIA installation path where the profile gets created under *PS_HOME\webserv* during the normal PIA install. Keep the *Nodename* and *Cellname* with default values.

8. On *WebSphere1*, add the node to DM (assuming DM is also installed on WebSphere 1) using the following command:

```
Windows:

PS_HOME\webserv\<appname>\bin\addNode.bat
<DeploymentManager_host> <8879>
```

```
Unix:
PS_HOME/webserv/<appname>/bin/addNode.sh
<DeploymentManager_host> <8879>
```

where *<DeploymentManager_host>* is the DeploymentManager hostname and *<8879>* is the SOAP connector port for example. This will stop any servers currently running on the node.

Note: DM's SOAP port is listed in *AboutThisProfile.txt* located in *DeploymentManager_Home\logs*.

9. The *NodeManager* will be started as part of the *addNode* command. If not, start the NodeManager using:

```
Windows: PS_HOME\webserv\<appname>\bin\startNode.bat

Unix: PS_HOME/webserv/<appname>/bin/startNode.sh
```

10. On the rest of the WebSphere instances *Websphere2……WebSphere N* where you have installed customer profiles, run the *addNode* command to add node to DM.

11. Open a browser and access the *WebSphere Administrative Console* pointing to the *DeploymentManager_host* machine

```
eg. http://<DeploymentManager_host>:9060/ibm/console
```

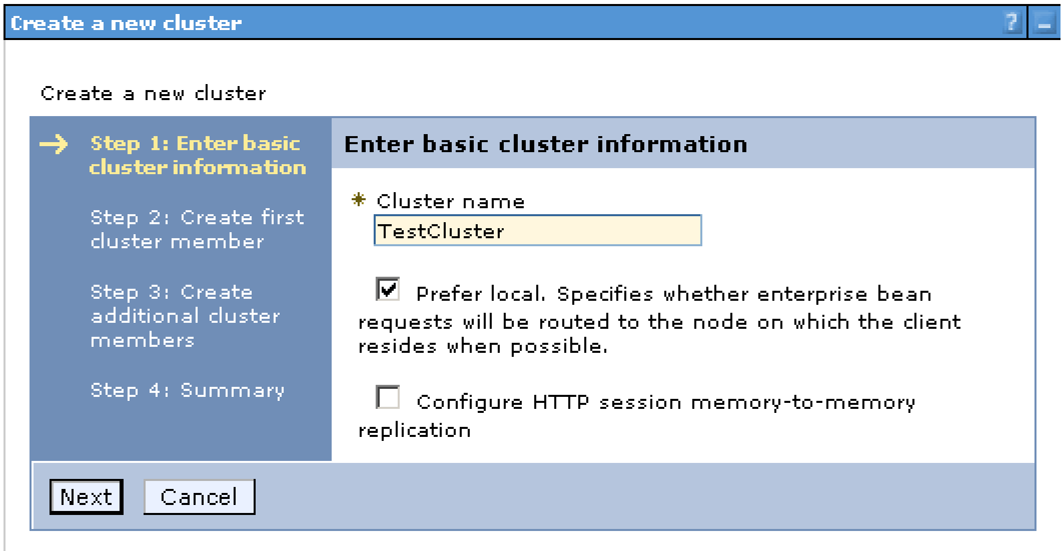
Note: DM's admin console port is listed in *AboutThisProfile.txt* located in *DeploymentManager_Home\logs*.

12. **Create a Cluster**

    a. Select *Servers -> Clusters*

    b. Click *New* to create a new cluster.

c. Call the new cluster *PeopleSoftCluster*. *Select Create the server using an existing server as a template* and choose the *server1* that came with the PIA install on *WebSphere 1* instance. Click *Next*



Create a new cluster
Create a new cluster
Step 1: Enter basic cluster information
Step 2: Create first cluster member
Step 3: Create additional cluster members
Step 4: Summary
Next
Cancel

Enter basic cluster information

* Cluster name: TestCluster

☑ Prefer local. Specifies whether enterprise bean requests will be routed to the node on which the client resides when possible.

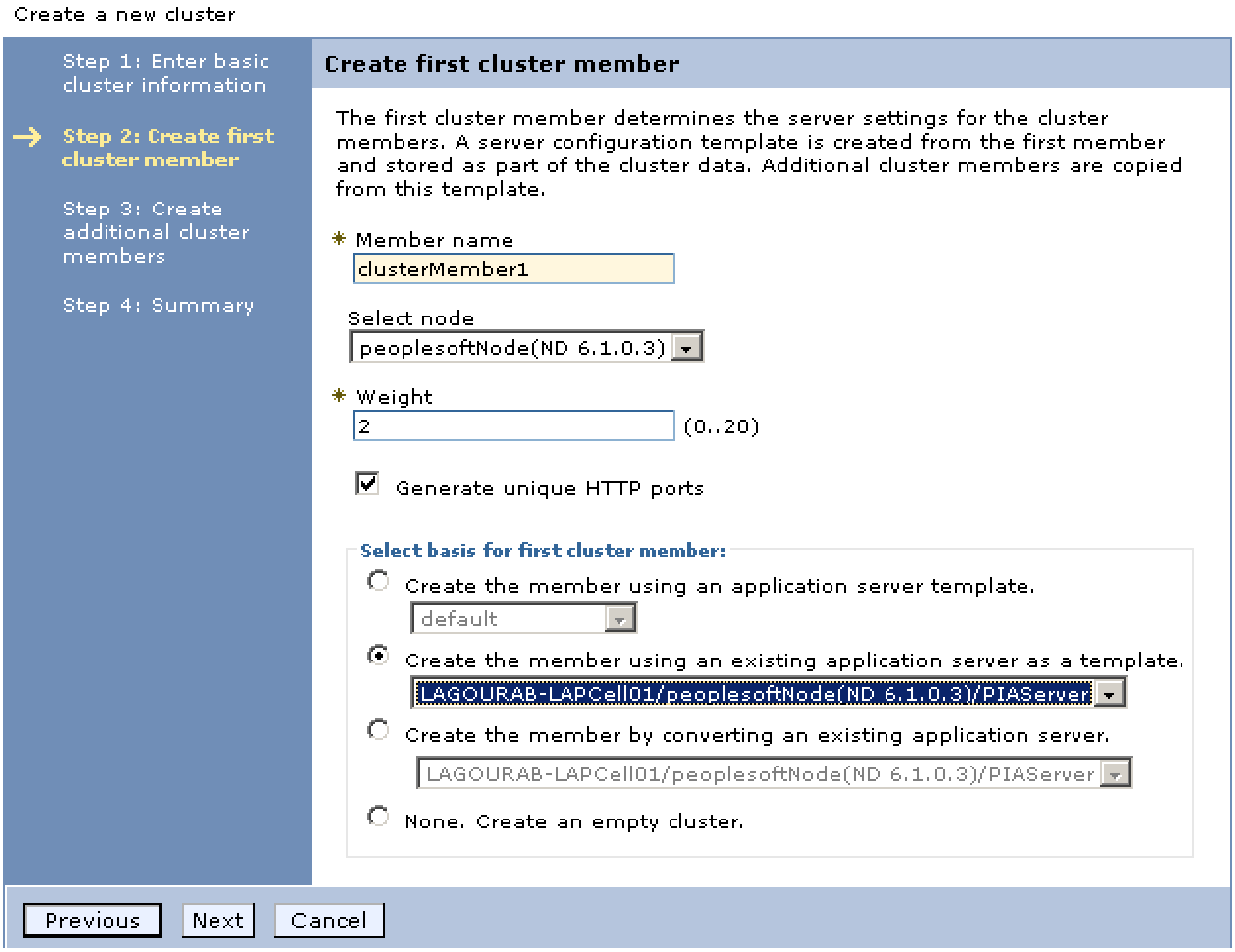
☐ Configure HTTP session memory-to-memory replication

d. Add one or more cluster members (depending on the number of application servers) to this cluster.

i. Enter a *Name* for the new cluster member (eg. *ClusterMember1*).

ii. From the drop down list, select the *Node* that the cluster member is to be associated with. Remember that a cluster can be deployed across multiple nodes with multiple cluster members on each node. If using different nodes, make sure to select a different node for the cluster members.

iii. Select the checkbox to *Generate Unique HTTP Ports*. WebSphere will assign unique ports to the new cluster member. These ports should be double checked to prevent interference with other applications' ports. Click *Next*



Create a new cluster

Step 1: Enter basic cluster information
→ Step 2: Create first cluster member
Step 3: Create additional cluster members
Step 4: Summary

Create first cluster member

The first cluster member determines the server settings for the cluster members. A server configuration template is created from the first member and stored as part of the cluster data. Additional cluster members are copied from this template.

* Member name: clusterMember1

Select node: peoplesoftNode(ND 6.1.0.3)

* Weight: 2 (0..20)

☑ Generate unique HTTP ports

Select basis for first cluster member:

○ Create the member using an application server template.
default

● Create the member using an existing application server as a template.
LAGOURAB-LAPCell01/peoplesoftNode(ND 6.1.0.3)/PIAServer

○ Create the member by converting an existing application server.
LAGOURAB-LAPCell01/peoplesoftNode(ND 6.1.0.3)/PIAServer

○ None. Create an empty cluster.

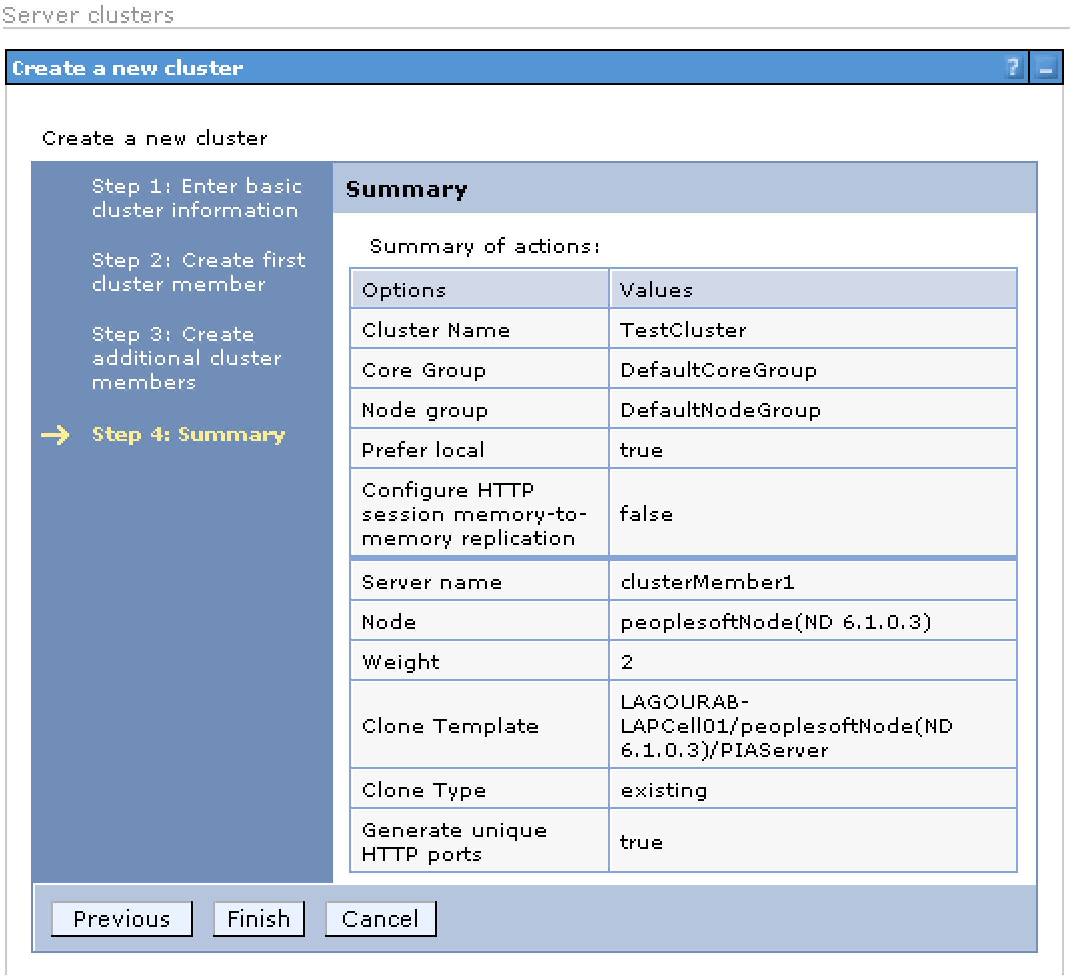
iv. The new cluster member will appear in the application server list.

v. Repeat steps i-iv to add one or more cluster members on each node. For best performance, balance the cluster members across multiple nodes.

vi. When finished adding cluster members, click *Next*.

vii. A summary of the changes will appear.

viii. Click *Finish* to create the cluster members.



Server clusters

Create a new cluster

Create a new cluster

Step 1: Enter basic cluster information
Step 2: Create first cluster member
Step 3: Create additional cluster members
→ Step 4: Summary

Summary

Summary of actions:

Options	Values
Cluster Name	TestCluster
Core Group	DefaultCoreGroup
Node group	DefaultNodeGroup
Prefer local	true
Configure HTTP session memory-to-memory replication	false
Server name	clusterMember1
Node	peoplesoftNode(ND 6.1.0.3)
Weight	2
Clone Template	LAGOURAB-LAPCell01/peoplesoftNode(ND 6.1.0.3)/PIAServer
Clone Type	existing
Generate unique HTTP ports	true

ix. You will see a warning message that changes have been made. You will need to save the changes by clicking the *Save* link and saving the configuration. Be sure to select *Synchronize changes with Nodes* to allow the nodes to be updated with changes.

x. On each cluster member (typically a server), make sure all the properties are set properly as mentioned in step 13.

13. Open a browser and access the *WebSphere Administrative Console* pointing to the *DeploymentManager* machine. If you have created servers based on the application server template of *WebSphere 1* instance, make sure the servers on *WebSphere 2* to *WebSphere N* and the *PeopleSoft* application has the custom properties set properly as listed below.

Note: If these properties or configuration is already set, do not change any settings. These steps are listed here to make sure that the PIA configuration is setup properly in the cluster setup.

- **Check Shared Library**

  - Goto *Environment->Shared Libraries* and click on *New*
  - Enter the name as *psSharedLibrary* and select the scope of the library to *Node* and *server1* (or the server you have created) in the AS profile
  - Enter the classpath details as given in this example (the example values are given assuming that you have installed PIA into *PS_HOME* as *D:/PT849* and entered the application name as *84903domain*)

```
D:/PT849/webserv/84903domain/installedApps/84903domainNodeCell/84903domain.ear/lib/pluto-1.0.1.jar

D:/PT849/webserv/84903domain/installedApps/84903domainNodeCell/84903domain.ear/lib/portlet-api-1.0.jar

D:/PT849/webserv/84903domain/installedApps/84903domainNodeCell/84903domain.ear/lib/saaj.jar

D:/PT849/webserv/84903domain/installedApps/84903domainNodeCell/84903domain.ear/lib/xalan.jar

D:/PT849/webserv/84903domain/installedApps/84903domainNodeCell/84903domain.ear/PSIGW.war/WEB-INF/lib/mail.jar

D:/PT849/webserv/84903domain/installedApps/84903domainNodeCell/84903domain.ear/PSIGW.war/WEB-INF/lib/activation.jar
```

After adding the library, associate this library with *server1* (or the server that you created) by doing the following actions,

- ✓ Go to *server1 -> Java and process management -> Class loader*
- ✓ Create a new *ClassLoader* and select *Classes loaded wih parent class loader first* as the order
- ✓ Select *Shared library references* under *Additional properties*
- ✓ Add a library by selecting the library from the drop

down list of libraries

- **Check Custom Properties**

  - Go to *Servers -> Applications Servers -> server1* (or the server that you created)

  - Go to *Java and Process Management* under *Server Infrastructure*, click on *Process definition*

  - Click on *Java Virtual Machine* under *Additional properties*

  - Click on *Custom Properties* under *Additional Properties*

  - Enter the following properties (the example values are given assuming that you have installed PIA into *PS_HOME* as *D:\PT849* and enter the application name as *84903domain*)

    - ✓ Set *HttpSessionIdReuse* property to *false*

    - ✓ Set *ps_vault* property to the path value like *D:/PT849/webserv/84903domain/installedApps/84903domainNodeCell/84903domain.ear/psvault*

    - ✓ Set *java.util.logging.config.file* property to the path value like *D:\PT849\webserv\84903domain\installedApps\84903domainNodeCell\84903domain.ear\logging.properties*

    - ✓ Set *java.util.logging.configureByLoggingPropertiesFile* property to *true*

  - Go to *Application servers -> server1 -> Web container -> Web container transport chains -> WCInboundDefault -> HTTP inbound channel (HTTP_2) -> Custom Properties*

    - ✓ Set *CookiesConfigureNoCache* property to the value *false*

  - Go to *Application servers -> server1 -> Web container -> Web container transport chains -> WCInboundDefaultSecure -> HTTP inbound channel (HTTP_4) > Custom Properties*

    - ✓ Set *CookiesConfigureNoCache* property to the value *false*

  - Go to *Application servers -> server1 -> Web container -> Custom Properties*

    - ✓ Set *CheckCookiesOnDispatch* property to the value *true*

- **Check Log Levels**
  - Go to *Application servers -> server1 -> Logging and Tracing -> Diagnostic Trace Service -> Change Log Detail Levels*
  - Change the logging level to severe (change **=info* to **=severe*) and click *Apply* button
- **Create Virtual Host entries**
  - Go to *Environment -> Virtual Hosts*
  - Select *default_host*.
  - Click on *Host Aliases* under *Additional Properties.* Make new entries for HTTP and HTTPS ports belonging to each of the cluster members. On each cluster member the HTTP and HTTPS ports are given by entries *WC_defaulthost* and *WC_defaulthost_secure* entries in the *Servers -> Application Servers -> ClusterMember -> Ports* screen
  - Select *Mime Types* under *Additional Properties*
  - Add a new mime type *image/gif* and extension as *GIF* and click *Apply* button.
- **Check ProfileName**
  - Go to *Servers -> Applications Servers* and click on *server1* (or the server that you created)
  - Click on *Java Process Management* under *Server Infrastructure*
  - Click on *Process Definition* and click *Java Virtual Machine* under *Additional Properties*
  - Find a parameter called *Generic JVM arguments* and enter the following value

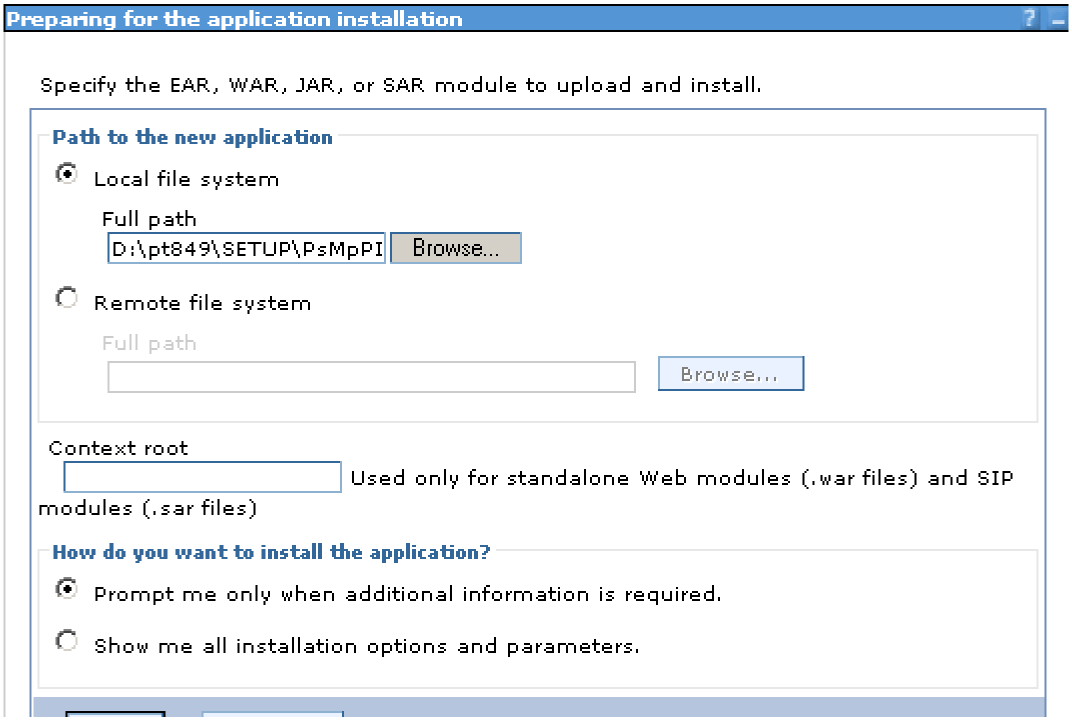
    -Dprofile.name=<yourprofilename>

    ```
    (for example -Dprofile.name=PeopleSoft)
    ```

14. Install the PeopleSoft PIA

- Select *Enterprise Applications -> Install New Application*.

- Select *Browse* and navigate to the local path containing the EAR file (the EAR file was created in step 6f). Click *Next*. This step will take sometime for EAR to get uploaded.
  ```
  Windows: C:\PS_HOME\peoplesoft.ear
  Unix: /PS_HOME/peoplesoft.ear
  ```



Preparing for the application installation

Specify the EAR, WAR, JAR, or SAR module to upload and install.

Path to the new application

● Local file system

Full path: D:\pt849\SETUP\PsMpPI

○ Remote file system

Full path:

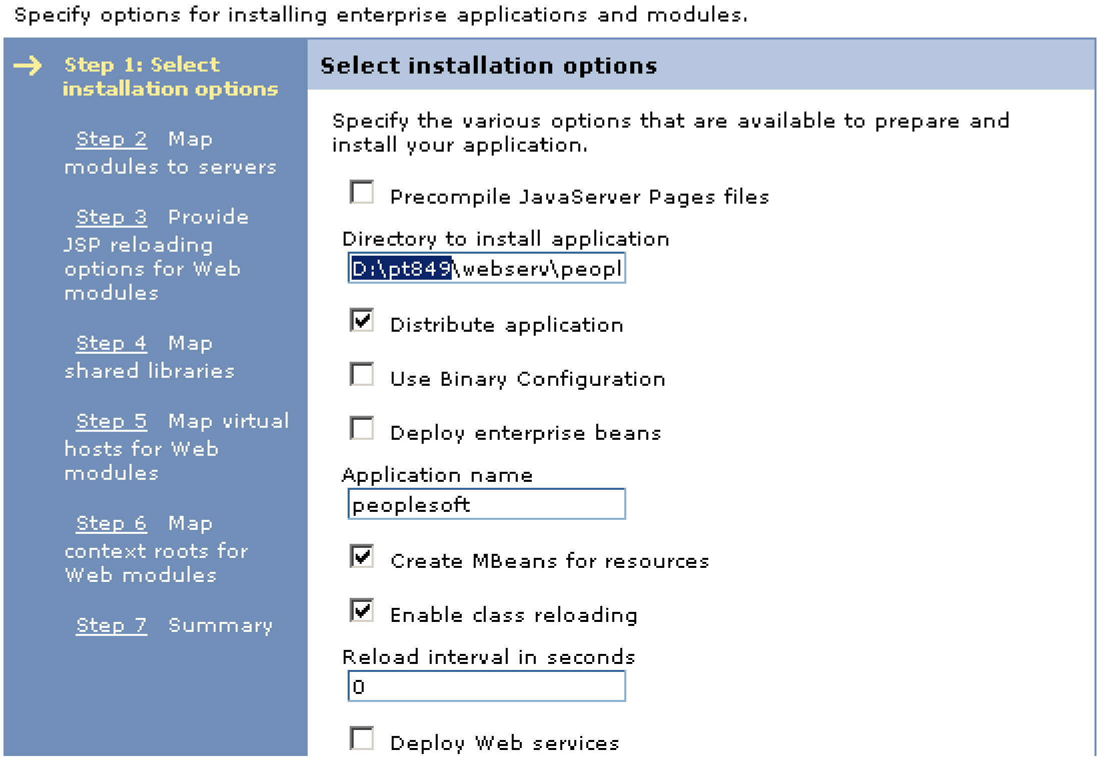
Context root: Used only for standalone Web modules (.war files) and SIP modules (.sar files)

How do you want to install the application?

● Prompt me only when additional information is required.

○ Show me all installation options and parameters.

- Using the values recorded in step 6b during the PIA install, set *Directory to Install Application* to *PS_HOME\webserv\<appname>\installedApps\<hostname>NodeCell* which was the directory that the PIA was installed to on the staging machine or *WebSphere 1* instance. This value must remain the same because the PIA contains hard coded references to directory structures. Change *Application Name* to the value specified in step 6b.



Specify options for installing enterprise applications and modules.

→ Step 1: Select installation options
Step 2 Map modules to servers
Step 3 Provide JSP reloading options for Web modules
Step 4 Map shared libraries
Step 5 Map virtual hosts for Web modules
Step 6 Map context roots for Web modules
Step 7 Summary

Select installation options

Specify the various options that are available to prepare and install your application.

☐ Precompile JavaServer Pages files

Directory to install application: D:\pt849\webserv\peopl

☑ Distribute application

☐ Use Binary Configuration

☐ Deploy enterprise beans

Application name: peoplesoft

☑ Create MBeans for resources

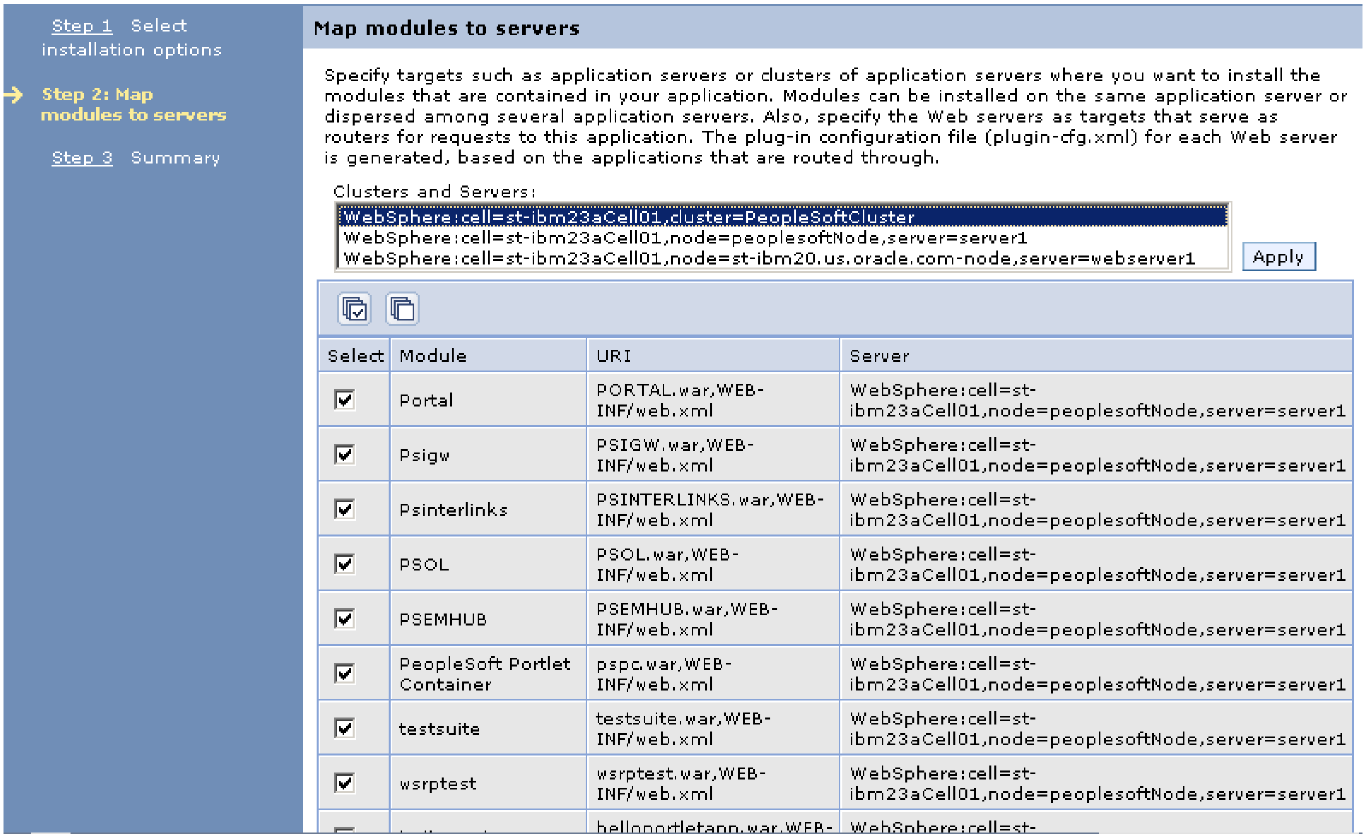
☑ Enable class reloading

Reload interval in seconds: 0

☐ Deploy Web services

- **Note**: If you want to enter any other *PS_HOME* and *Application Name*, then beware of making the following changes inorder for the application to work,
  - ✓ Open *web.xml* file on respective cluster members at the location *PS_HOME/webserv/<appname>/installedApps/<hostname> NodeCell/< application_name>/PORTAL* and change *<param-value>* in *<init-param>* to appropriate *PS_HOME*. Similarly change can be made to web.xml for *PSIGW* Web Module
  - ✓ Make sure to correct the library paths in *psSharedLibrary* in *Environment -> Shared Libraries* screen for each cluster member.
  - ✓ Make sure to correct the path values for JVM custom properties *java.util.logging.config.file* and *ps_vault* for each cluster member.
  - ✓ Inorder for certain utility scripts in the folder *PS_HOME/webserv/<appname>/installedApps/<hostname> NodeCell/< application_name>/* to work properly make sure to correct the path values in utility scripts like *pskeymanager.cmd (or .sh), HashKeyGenerator.bat (or .sh), StartSendMaster.bat (or .sh)* etc

- In *Map modules to application servers* check all the listed modules and select the *PeopleSoftCluster* in the *Clusters and Servers* listbox. Click *Apply*. The server mappings should change to match the cluster name. To verify, look in the third column titled *Server*. Click *Next*


Step 1 Select installation options
→ Step 2: Map modules to servers
Step 3 Summary
Apply

Map modules to servers

Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

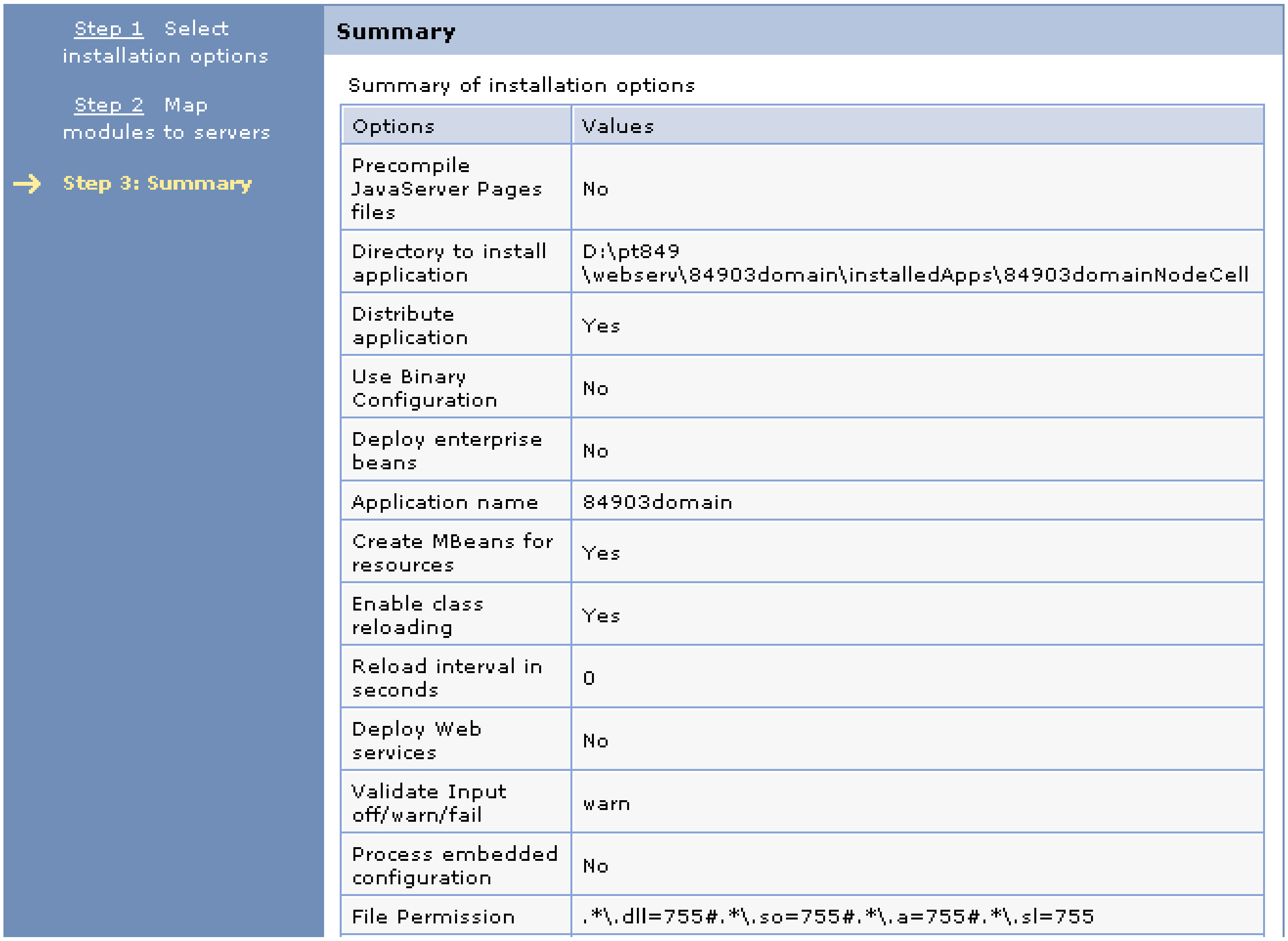
Clusters and Servers:

WebSphere:cell=st-ibm23aCell01,cluster=PeopleSoftCluster
WebSphere:cell=st-ibm23aCell01,node=peoplesoftNode,server=server1
WebSphere:cell=st-ibm23aCell01,node=st-ibm20.us.oracle.com-node,server=webserver1

Select	Module	URI	Server
☑	Portal	PORTAL.war,WEB-INF/web.xml	WebSphere:cell=st-ibm23aCell01,node=peoplesoftNode,server=server1
☑	Psigw	PSIGW.war,WEB-INF/web.xml	WebSphere:cell=st-ibm23aCell01,node=peoplesoftNode,server=server1
☑	Psinterlinks	PSINTERLINKS.war,WEB-INF/web.xml	WebSphere:cell=st-ibm23aCell01,node=peoplesoftNode,server=server1
☑	PSOL	PSOL.war,WEB-INF/web.xml	WebSphere:cell=st-ibm23aCell01,node=peoplesoftNode,server=server1
☑	PSEMHUB	PSEMHUB.war,WEB-INF/web.xml	WebSphere:cell=st-ibm23aCell01,node=peoplesoftNode,server=server1
☑	PeopleSoft Portlet Container	pspc.war,WEB-INF/web.xml	WebSphere:cell=st-ibm23aCell01,node=peoplesoftNode,server=server1
☑	testsuite	testsuite.war,WEB-INF/web.xml	WebSphere:cell=st-ibm23aCell01,node=peoplesoftNode,server=server1
☑	wsrptest	wsrptest.war,WEB-INF/web.xml	WebSphere:cell=st-ibm23aCell01,node=peoplesoftNode,server=server1
		helloportletapp.war,WEB-	WebSphere:cell=st-

- In *Map virtual hosts to Web modules* screen, accept the *default_host* as the *Virtual Host* for the *Web Modules*. No changes are required. Click *Next*.

- On the *Summary* screen, review your choices. If a problem is seen, click *Previous* to correct the error. Make sure that the *application name* and the *Directory to Install Application* match the values used in the PIA installation in step 6b. Click *Finish*. This step will take a few minutes.

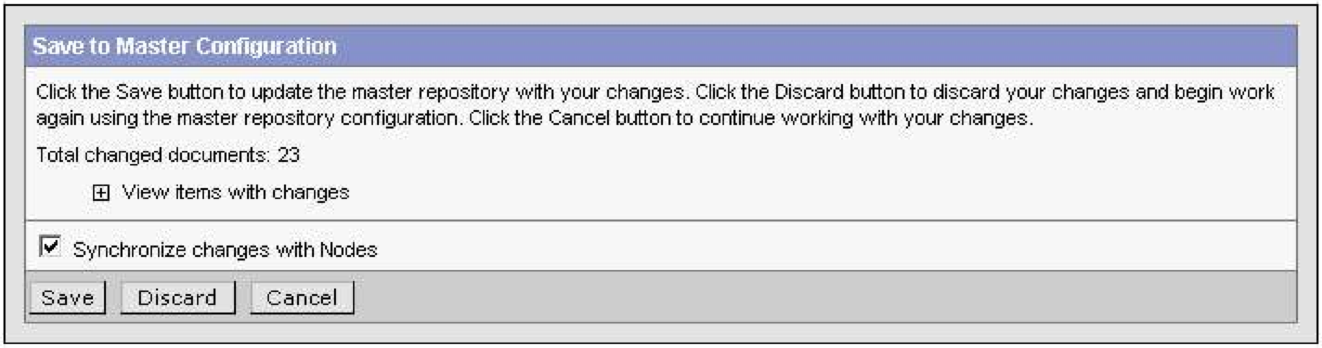

Step 1 Select installation options
Step 2 Map modules to servers
→ Step 3: Summary

Summary

Summary of installation options

Options	Values
Precompile JavaServer Pages files	No
Directory to install application	D:\pt849 \webserv\84903domain\installedApps\84903domainNodeCell
Distribute application	Yes
Use Binary Configuration	No
Deploy enterprise beans	No
Application name	84903domain
Create MBeans for resources	Yes
Enable class reloading	Yes
Reload interval in seconds	0
Deploy Web services	No
Validate Input off/warn/fail	warn
Process embedded configuration	No
File Permission	.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755

- Select option *Save to Master Configuration*. You will see a page that displays the status of the *PeopleSoft* application install as successful. Check the option *Synchronize changes with Nodes*. Click *Save* button. This step will take a few minutes to copy the application to all the nodes. The Configuration for all servers will be updated, and the EAR file will be transferred and unpacked on all attached servers.



Save to Master Configuration

Click the Save button to update the master repository with your changes. Click the Discard button to discard your changes and begin work again using the master repository configuration. Click the Cancel button to continue working with your changes.

Total changed documents: 23

⊞ View items with changes

☑ Synchronize changes with Nodes

- **Renaming JSESSIONID**

  ✓ Go to *Enterprise Applications -> PeopleSoft* application and select *Manage Modules* link

  ✓ Click on *Portal.war*

  ✓ Click on *Session Management* under *Additional properties*. Click *Enable cookies* link and set the *Cookie Name* to be of the format as *<hostname>-<PIAport>-PORTAL-PSJSESSIONID* (for example *PTA107-10001-PORTAL-PSJSESSIONID*)

  ✓ If you have entered *AuthtokenDomain* during the PIA installation enter the same for *Cookie domain* (for example ***.**peoplesoft**.**com*) and click *Apply*

  ✓ Click on *Session Management* under *Additional properties* and make sure to select the check box for *Override session management* (ie., enable this option)

  ✓ Click on *PeopleSoft Portlet Container (PSPC.war)* module from the table (when you click on *Manage Modules*)

  ✓ Follow the same steps for *PSPC.war* module and set the *Cookie Name* to be of the format as *<hostname>-<PIAport>-PORTLET-PSJSESSIONID* (for example *PTA107-10001- PORTLET - PSJSESSIONID*)

  ✓ If you have entered *AuthtokenDomain* during the PIA installation enter the same for *Cookie domain* (for example ***.**peoplesoft**.**com* and Click *Apply*

- Make sure to save all changes made to the configuration by selecting *Save changes to master repository* from under *System Administration* menu. Before saving make sure any change made to configuration gets propagated to all cluster nodes by enabling the option *Synchronize changes with Nodes* in *Console Preferences* screen.

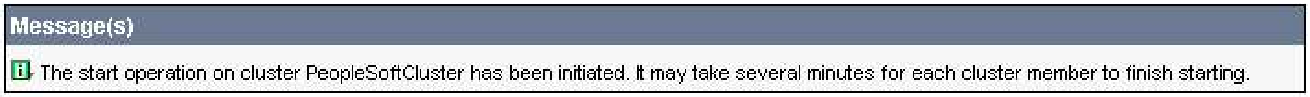
15. If you have configured *IBM HTTP Server* (or any other webserver like *Sun Java or IIS*) as Reverse Proxy in front of the Deployment Manager, make sure to add the HTTP and HTTPS port entries of the RPS server in virtual host *default_host*. Then generate the webserver plugin *plugin-cfg.xml* file and propagate it to the location where you have created the RPS webserver definition.

16. Start the Cluster

i. Select *Servers->Clusters*

ii. Check *PeopleSoftCluster*

iii. Click *Start*

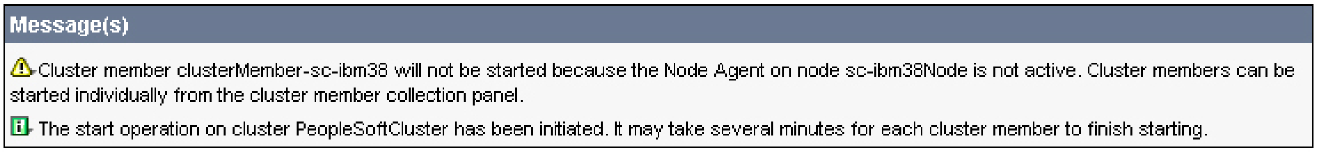


Message(s)

The start operation on cluster PeopleSoftCluster has been initiated. It may take several minutes for each cluster member to finish starting.

iv. When all of the servers in the cluster have started the status icon will update to a solid green arrow.

v. If a message appears warning that a cluster member will not be started, check that the node agent is running on that machine (use the *startNode.bat (or .sh)* command located in *<PS_HOME>/webserv/<appname>/bin*). Once that node agent is started, select the cluster and click *Start* to start the cluster member.



Message(s)

Cluster member clusterMember-sc-ibm38 will not be started because the Node Agent on node sc-ibm38Node is not active. Cluster members can be started individually from the cluster member collection panel.

The start operation on cluster PeopleSoftCluster has been initiated. It may take several minutes for each cluster member to finish starting.

Additional Resources

- **WebSphere Network Deployment V6.1 Info Center**

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/welcome_nd.html

- **WebSphere V6.1 Supported Platforms** http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg27007651

- **WebSphere Redbooks** (http://www.redbooks.ibm.com)

WebSphere Application Server V6.1 System Management & Configuration (SG24-7304-00)

WebSphere Application Server V6.1: Security Handbook (SG24-6316-01)

WebSphere Application Server V6: Scalability and Performance Handbook (SG24-6392-00)

CONFIGURING A WEBSPHERE CLUSTER WITH PEOPLETOOLS 8.50

WebSphere 7.0 is the certified webserver for PT 8.50 release. The steps to create a cluster with WebSphere 7.0 remains largely similar to WebSphere 6.1 barring a few changes in the screenshots. There are also some minor changes in the way PeopleSoft is deployed in PT 8.50 release as opposed to PT 8.49 release.

Configuring a WebSphere Cluster using WAS ND 7.0

1. Install WebSphere Application Server ND 7.0 on multiple servers(*WebSphere 1* to *WebSphere N*) as per your requirement. Use the instructions provided in the *PeopleTools 8.50 Install Guide*.

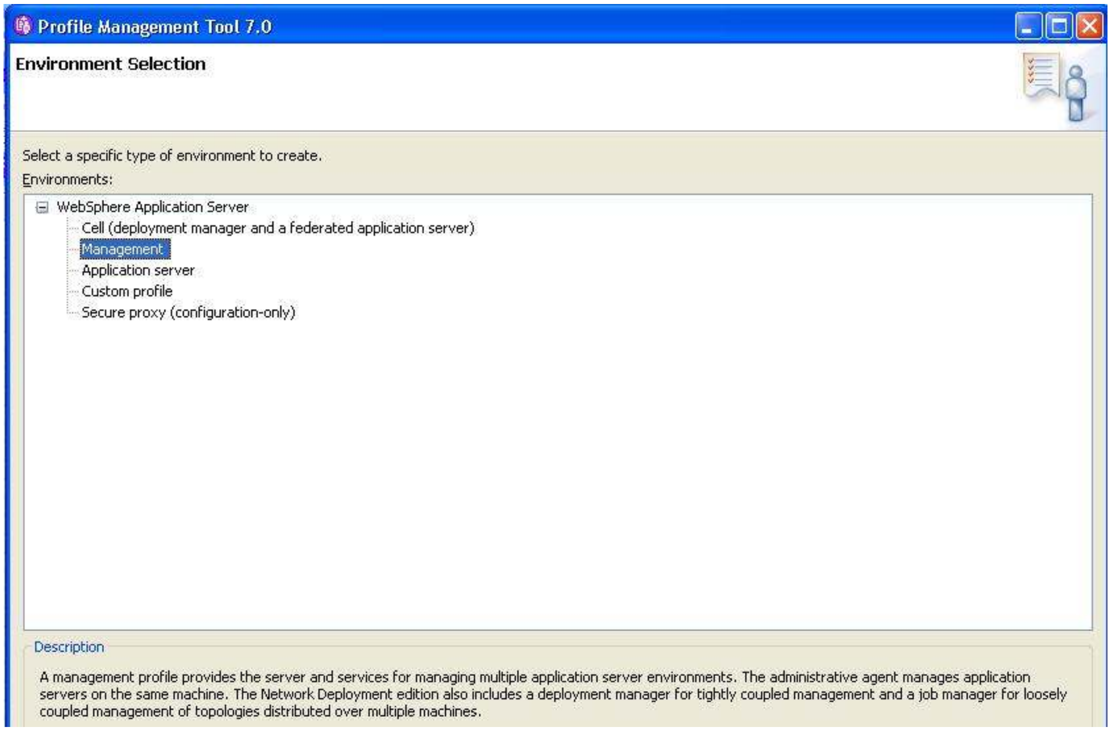
2. Install PIA from PT 8.50 and select Single Server Installation on one of the WebSphere servers (*WebSphere1).* Use the instructions provided in the *PeopleTools 8.50 Install Guide*.

3. Select one of the servers to host the *Deployment Manager* from the available WebSphere servers (*WebSphere1, WebSphere 2…..WebSphere N*).

4. On the machine which will host DM (Deployment Manager), assume that it is server *WebSphere1*, use *Profile Management Tool* and create a *Deployment Manager* profile. Follow the steps listed below to create the DM profile,

    - Go to *WAS_HOME\bin\ProfileManagement\bin\* and run *PMT.bat (or .sh)* and this will bring up a profile creation tool wizard.

    - Select *Management* profile type and select *Deployment Manager* server type under it.



Environment Selection

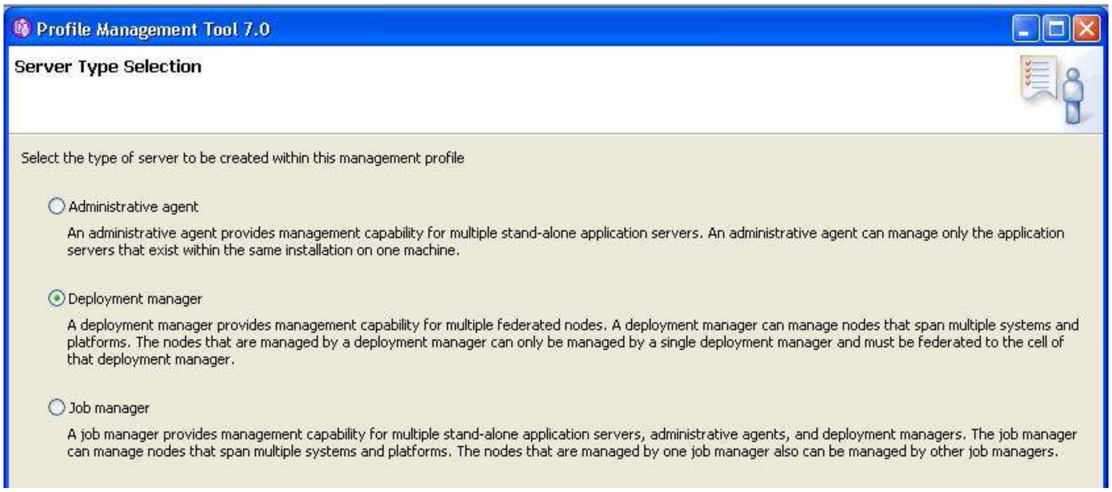
Select a specific type of environment to create.

Environments:

- WebSphere Application Server
  - Cell (deployment manager and a federated application server)
  - Management
  - Application server
  - Custom profile
  - Secure proxy (configuration-only)

Description

A management profile provides the server and services for managing multiple application server environments. The administrative agent manages application servers on the same machine. The Network Deployment edition also includes a deployment manager for tightly coupled management and a job manager for loosely coupled management of topologies distributed over multiple machines.



Server Type Selection

Select the type of server to be created within this management profile

○ Administrative agent

An administrative agent provides management capability for multiple stand-alone application servers. An administrative agent can manage only the application servers that exist within the same installation on one machine.

● Deployment manager

A deployment manager provides management capability for multiple federated nodes. A deployment manager can manage nodes that span multiple systems and platforms. The nodes that are managed by a deployment manager can only be managed by a single deployment manager and must be federated to the cell of that deployment manager.

○ Job manager

A job manager provides management capability for multiple stand-alone application servers, administrative agents, and deployment managers. The job manager can manage nodes that span multiple systems and platforms. The nodes that are managed by one job manager also can be managed by other job managers.

- Select *Advanced Profile creation* option and click *Next*
- Select *Deploy the administrative console* option and click *Next*
- Select a name for profile and the location where the profile will be created and click *Next*
- Keep the node name, cell name and hostname same and click *Next*
- Uncheck the *Enable administrative security* option and click *Next*
- Keep the port values as assigned and click *Next*
- Uncheck the *Run Deployment Manager as windows service* option and click *Next*

- Verify all the items on the Profile Creation summary page and click *create* button

This will create a *Deployment manager* profile and you can use it setup the cluster.

5. If using a webserver (like *IBM HTTP Server, Sun Java Webserver or IIS*) as *Reverse Proxy Server* in front of DM, install a supported webserver on *HTTPServer* machine. Use the instructions provided in the *PeopleTools 8.50 Install guide*.

6. On *WebSphere1*, install the PIA as mentioned in step 2, record settings.

   a. Install PIA using the instructions in the *PeopleTools 8.50 Install guide*.

   b. Record *PS_HOME, AuthTokenDomain, Application Name* and *psreports* path. This information is very important and should be saved for future reference.

      Note: The value for PS_HOME will be used by all machines in the cluster, so make sure the path is valid for each machine. The same values of PS_HOME and application name need to be used when custom profiles are created on *WebSphere 2* to *WebSphere N* instances

   c. Create any additional PeopleSoft sites.

   d. Perform any required customizations.

   e. Test to ensure setup works.

   f. Execute WebSphere's *EARExpander* from *PS_HOME/webserv/<appname>/bin* to create an EAR file with the customizations.

```
Windows:

C:\<PS_HOME>\webserv\<appname>\bin\EARExpander.
bat –ear C:\<PS_HOME>\peoplesoft.ear –
operationDir
C:\<PS_HOME>\webserv\<appname>\installedApps\<h
ostname>NodeCell\<appname>.ear –operation
collapse

Unix:
/<PS_HOME>/
webserv/<appname>/bin/EARExpander.sh –ear
/<PS_HOME>/peoplesoft.ear –operationDir
/<PS_HOME>/webserv/<appname>/installedApps/<hos
tname>NodeCell/<appname>.ear –operation
collapse
```

Be sure the Deployment Manager is started on the machine where you installed DM profile

```
Windows: DeploymentManager_Home\bin\startManager.bat
```

```
Unix: DeploymentManager_Home/bin/startManager.sh
```

Where *DeploymentManager_Home* is the location of the DM profile that you have created on step 4.

7. On *WebSphere2* to *WebSphereN,* create *Custom* profiles and add these nodes to DM.

    a. Before federating the node, check that the date and times for the machines are within 5 minutes of each other. If the machines do not have similar times, an error message will appear and the node will not be added.

    b. The *Deployment Manager* machine and the *Node Agent* machine must be able to resolve the hostnames of each other. To test, use the hostname with the ping command.

    c. Create *Custom* profiles on machines *WebSphere 2* to *WebSphere N* using the *Profile Management Tool* as described in step 4, but select *Custom Profile* under *Environment Selection*. Enter the profilename same as the application name you have given during step 6b and set the profile path same as the PIA installation path where the profile gets created under *PS_HOME\webserv* during the normal PIA install. Keep the Nodename and Cellname with default values.

8. On *WebSphere1*, add the node to DM (assuming DM is also installed on WebSphere 1) using the following command:

```
Windows:

PS_HOME\webserv\<appname>\bin\addNode.bat
<DeploymentManager_host> <8879>

Unix:
PS_HOME/webserv/<appname>/bin/addNode.sh
<DeploymentManager_host> <8879>
```

where *<DeploymentManager_host>* is the DeploymentManager hostname and *<8879>* is the SOAP connector port for example. This will stop any servers currently running on the node.

Note: DM's SOAP port is listed in *AboutThisProfile.txt* located in *DeploymentManager_Home\logs*.

9. The *NodeManager* will be started as part of the *addNode* command. If not, start the *NodeManager* using:

```
Windows: PS_HOME\webserv\<appname>\bin\startNode.bat
Unix: PS_HOME/webserv/<appname>/bin/startNode.sh
```

10. On the rest of the WebSphere instances *Websphere2……WebSphere N* where you have installed customer profiles, run the *addnode* command to add node to DM.

11. Open a browser and access the WebSphere Administrative Console pointing to the *DeploymentManager_host* machine

```
e.g. http://<DeploymentManager_host>:9060/ibm/console
```

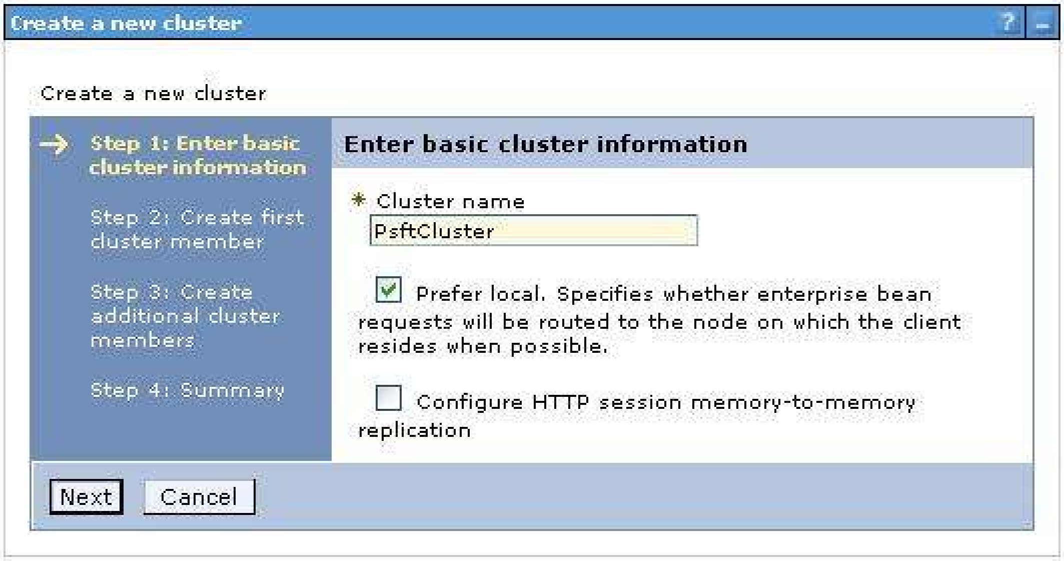
Note: DM's admin console port is listed in *AboutThisProfile.txt* located in *DeploymentManager_Home\logs*.

12. **Create a cluster**

    a. Select *Servers -> Clusters -> WebSphere application server clusters*

    b. Click *New* to create a cluster.

    c. Call the new cluster *PsftCluster*. Select option *Create the member using an existing application server as a template* and choose the *server1* that came with the PIA install on *WebSphere 1* instance. Click *Next*.



Create a new cluster
? –
Create a new cluster
→ Step 1: Enter basic cluster information
Step 2: Create first cluster member
Step 3: Create additional cluster members
Step 4: Summary
Enter basic cluster information
Next
Cancel

* Cluster name: PsftCluster

☑ Prefer local. Specifies whether enterprise bean requests will be routed to the node on which the client resides when possible.

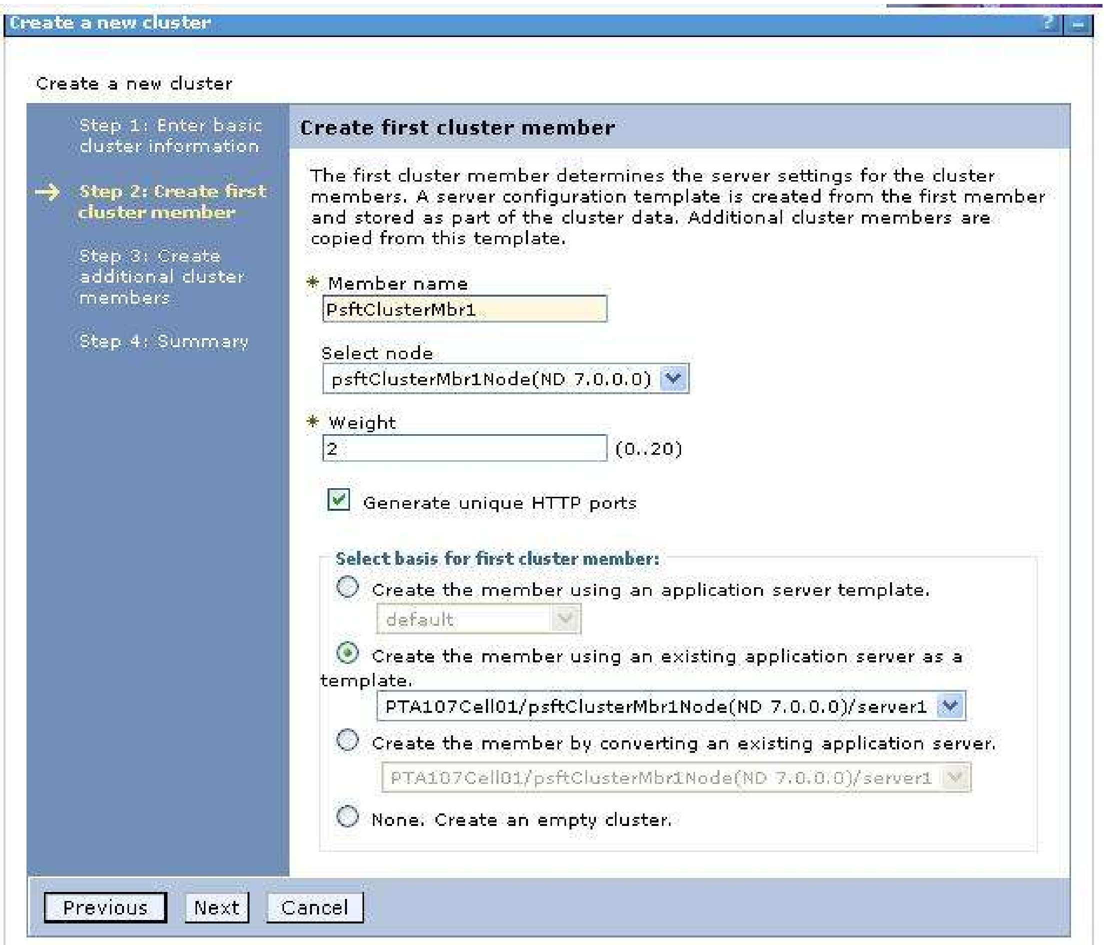
□ Configure HTTP session memory-to-memory replication

    d. Add one or more cluster members (depending on the number of application servers) to this cluster.

        i. Enter a Name for the new cluster member (ex. *PsftClusterMbr1*).

        ii. From the drop down list, select the *Node* that the cluster member is to be associated with. Remember that a cluster can be deployed across multiple nodes with multiple cluster members on each node. If using different nodes, make sure to select a different node for the cluster members.

iii. Select the checkbox to *Generate unique HTTP ports*. WebSphere will assign unique ports to the new cluster member. These ports should be double checked to prevent interference with other applications' ports. Click *Next.*



Create a new cluster

Create a new cluster

- Step 1: Enter basic cluster information
- → **Step 2: Create first cluster member**
- Step 3: Create additional cluster members
- Step 4: Summary

Create first cluster member

The first cluster member determines the server settings for the cluster members. A server configuration template is created from the first member and stored as part of the cluster data. Additional cluster members are copied from this template.

* Member name: PsftClusterMbr1

Select node: psftClusterMbr1Node(ND 7.0.0.0)

* Weight: 2 (0..20)

☑ Generate unique HTTP ports

Select basis for first cluster member:

○ Create the member using an application server template.
default

● Create the member using an existing application server as a template.
PTA107Cell01/psftClusterMbr1Node(ND 7.0.0.0)/server1

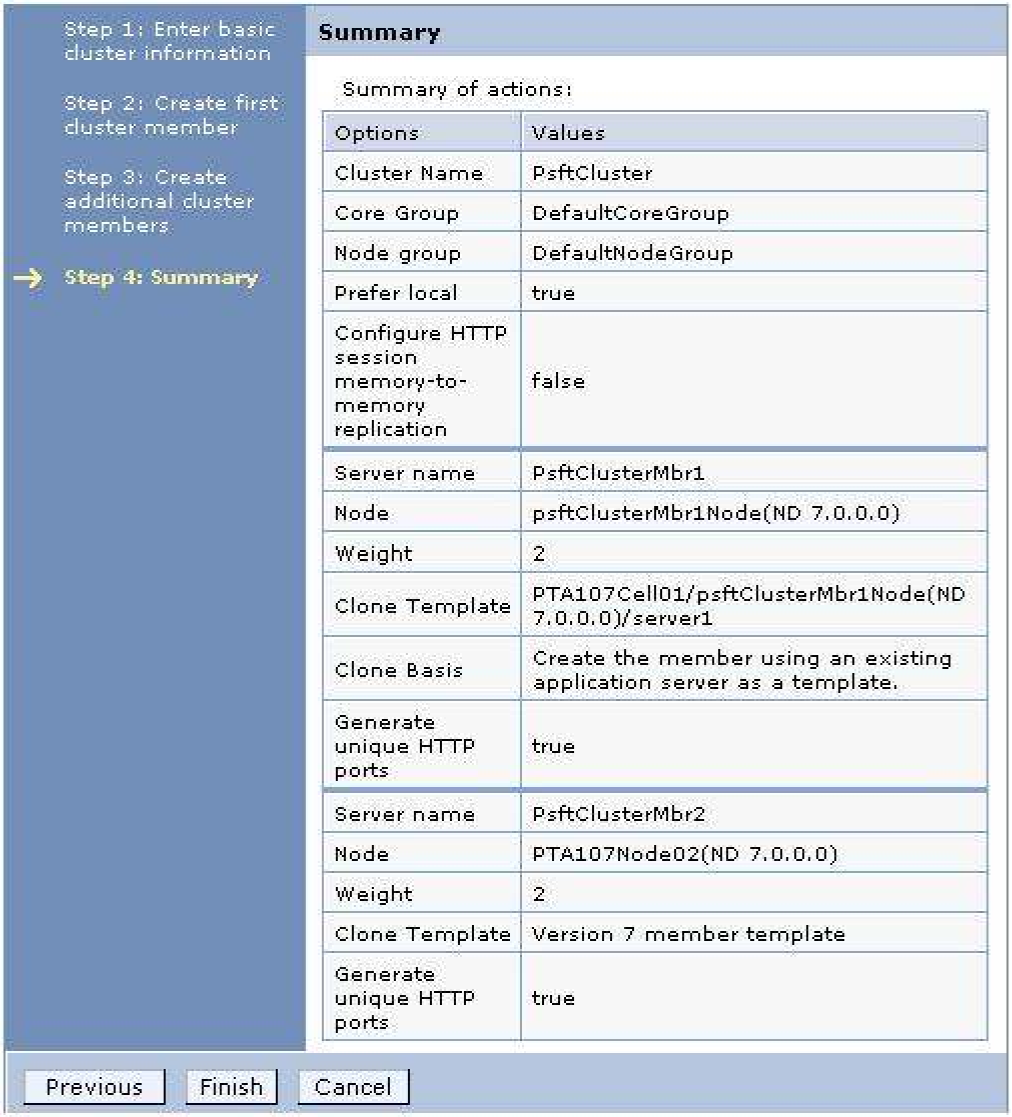
○ Create the member by converting an existing application server.
PTA107Cell01/psftClusterMbr1Node(ND 7.0.0.0)/server1

○ None. Create an empty cluster.

iv. The new cluster member will appear in the application server list.

v. Repeat steps i-iv to add one or more cluster members on each node. For best performance, balance the cluster members across multiple nodes. When finished adding cluster members, click *Next*.

vi. A summary of the changes will appear. Click *Finish* to create the cluster members.


Step 1: Enter basic cluster information
Step 2: Create first cluster member
Step 3: Create additional cluster members
Step 4: Summary
Previous
Finish
Cancel

Summary

Summary of actions:

Options	Values
Cluster Name	PsftCluster
Core Group	DefaultCoreGroup
Node group	DefaultNodeGroup
Prefer local	true
Configure HTTP session memory-to-memory replication	false
Server name	PsftClusterMbr1
Node	psftClusterMbr1Node(ND 7.0.0.0)
Weight	2
Clone Template	PTA107Cell01/psftClusterMbr1Node(ND 7.0.0.0)/server1
Clone Basis	Create the member using an existing application server as a template.
Generate unique HTTP ports	true
Server name	PsftClusterMbr2
Node	PTA107Node02(ND 7.0.0.0)
Weight	2
Clone Template	Version 7 member template
Generate unique HTTP ports	true

vii. You will see a warning message that changes have been made. You will need to save the changes by clicking the Save link and saving the configuration. Be sure to select *“Synchronize changes with Nodes”* to allow the nodes to be updated with changes.

viii. On each cluster member (typically a server), make sure all the properties are set properly as mentioned in step 13.

13. Open a browser and access the *WebSphere Administrative Console* pointing to the *DeploymentManager_host* machine.

```
eg. http://<DeploymentManager_host>:9060/ibm/console
```

If you have created servers based on the application server template of *WebSphere 1* instance, make sure the servers on *WebSphere 2* to *WebSphere N* and the *peoplesoft* application has the custom properties set properly as listed below.

Note: If these properties or configuration is already set, do not change any settings. These steps are listed here to make sure that the PIA configuration is setup properly in the Cluster setup.

- **Check shared library**

  - Go to *Environment -> Shared Libraries* and click on *New*
  - Enter the name as *psSharedLibrary* and select the scope of the library to *Node* and *server1* (or the server you have created) in the AS profile
  - Enter the classpath details as given in this example (the example values are given assuming that you have installed PIA into *PS_HOME* as *C:/PT8.50* and entered the application name as *psftClusterMbr1*)

```
C:/PT8.50/webserv/psftClusterMbr1/installedApps/psftClusterMbr1NodeCell/psftClusterMbr1.ear/PSIGW.war/WEB-INF/lib/mail.jar

C:/PT8.50/webserv/psftClusterMbr1/installedApps/psftClusterMbr1NodeCell/psftClusterMbr1.ear/PSIGW.war/WEB-INF/lib/activation.jar

C:/PT8.50/webserv/psftClusterMbr1/installedApps/psftClusterMbr1NodeCell/psftClusterMbr1.ear/lib/pluto-1.0.1.jar

C:/PT8.50/webserv/psftClusterMbr1/installedApps/psftClusterMbr1NodeCell/psftClusterMbr1.ear/lib/portlet-api-1.0.jar

C:/PT8.50/webserv/psftClusterMbr1/installedApps/psftClusterMbr1NodeCell/psftClusterMbr1.ear/lib/saaj.jar

C:/PT8.50/webserv/psftClusterMbr1/installedApps/psftClusterMbr1NodeCell/psftClusterMbr1.ear/lib/xalan.jar
```

After adding the library, associate this library with *server1*(or the server that you created) by doing the following actions,

- ✓ Go to *Server1 -> Java and process managemenet -> class loader*
- ✓ Create a new *ClassLoader* and select *Classes loaded wih parent class loader first* as the order
- ✓ Select *Shared library references* under *Additional properties*

  - ✓ Add a library by selecting the library from the drop down list of libraries

- **Create new Virtual Host entries**

  - Go to *Environment -> Virtual Hosts*. Create new virtual hosts namely *pia_host, psol_host* and *psemhub_host*.

  - Select *pia_host*.

  - Click on *Host Aliases* under *Additional Properties.* Make new entries for HTTP and HTTPS ports belonging to each of the cluster members. On each cluster member the HTTP and HTTPS ports are given by entries *WC_defaulthost* and *WC_defaulthost_secure* entries in the *Servers -> Application Servers -> ClusterMember -> Ports* screen

    - Click on *Mime Types* under *Additional Properties*. Add a new mime type entry *image/gif* and extension as *GIF*

    - Repeat the same by adding new *Host Alias* and *Mime Type* entries under other virtual hosts like *psol_host* and *psemhub_host* as well.

- **Check Custom Properties**

  - Go to *Servers -> Applications Servers -> server1* (or the server that you created)

  - Go to *Java and Process Management* under *Server Infrastructure*, click on *Process Definition*

  - Click on *Java Virtual Machine* under *Additional properties*

  - Click on *Custom Properties* under *Additional Properties*

  - Enter the following properties (the example values are given assuming that you have installed PIA into *PS_HOME* as *C:/PT8.50* and entered the application name as *psftClusterMbr1*)

    - ✓ Set *HttpSessionIdReuse* property to *false*

    - ✓ Set *java.util.logging.config.file* property to the path value like *C:/PT8.50/webserv/psftClusterMbr1/installedApps/psftClusterMbr1NodeCell/psftClusterMbr1.ear/logging.properties*

  - ✓ Set *java.util.logging.configureByLoggingPropertiesFile* property to *true*
  - ✓ Set *ps_vault* property to path value like *C:/PT8.50/webserv/psftClusterMbr1/installedApps/psftClusterMbr1NodeCell/psftClusterMbr1.ear/psvault*

- Go to *Application servers -> server1 -> Web container -> Web container transport chains -> WCInboundDefault -> HTTP inbound channel (HTTP_2) -> Custom Properties*
  - ✓ Set *CookiesConfigureNoCache* property to the value *false*
- Go to *Application servers -> server1 -> Web container -> Web container transport chains -> WCInboundDefaultSecure -> HTTP inbound channel (HTTP_4) -> Custom Properties*
  - ✓ Set *CookiesConfigureNoCache* property to the value *false*
- Go to *Application servers -> server1 -> Web container -> Custom Properties*
  - ✓ Set *CheckCookiesOnDispatch* property to the value *true*
  - ✓ Set *HttpSessionIdLength* property to the value *32*
- Go to *Application servers -> server1 -> Web container*. Select *pia_host* from the *Default virtual host* drop down list box.

➢ **Check Log Levels**

- Go to *Application servers -> server1 -> Logging and Tracing -> Diagnostic Trace Service -> Change Log Detail Levels*
- Change the logging level to *severe* (change **=info* to **=severe*) and click *Apply* button

➢ **Check ProfileName**

- Go to *Servers -> Applications Servers* and click on *server1* (or the server that you created)
- Click on *Java Process Management* under *Server Infrastructure*

- Click on *Process Definition* and click *Java Virtual Machine* under *Additional Properties*

- Find a parameter called *Generic JVM arguments* and enter the following value

  -Dprofile.name=<yourprofilename>

  ```
  (for example -Dprofile.name=PeopleSoft)
  ```

14. Install the PeopleSoft PIA

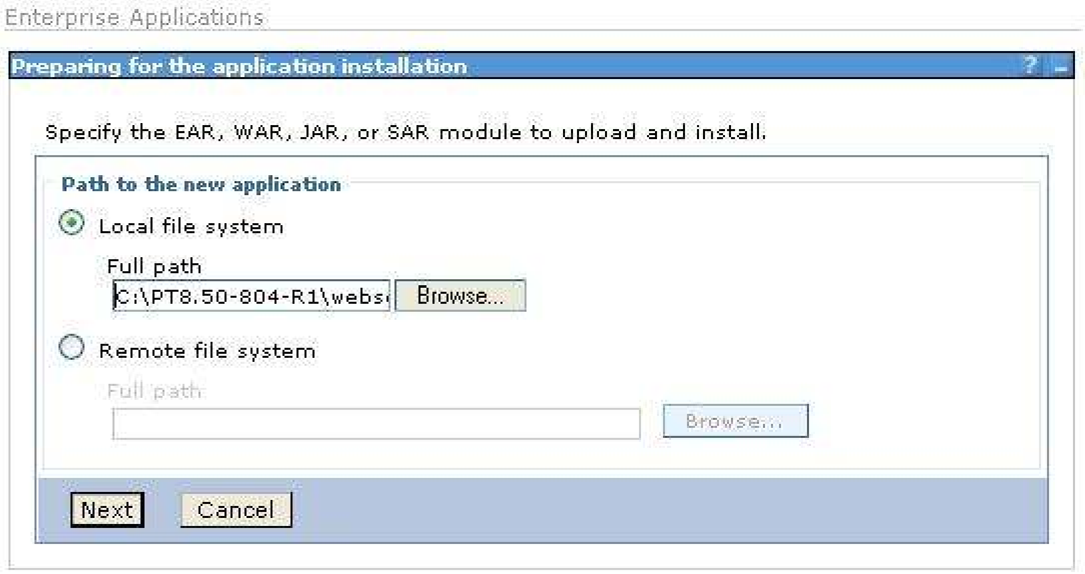
- Select *Enterprise Applications -> Install New Application*

- Select *Browse* and navigate to the local path containing the EAR file (the EAR file was created in step 6f). Select the EAR file.

  ```
  Windows: C:\<PS_HOME>\peoplesoft.ear
  ```

  ```
  Unix: /<PS_HOME>/peoplesoft.ear
  ```

  Click *Next*. This step will take sometime as the ear file is uploaded to WebSphere



Enterprise Applications

Preparing for the application installation

Specify the EAR, WAR, JAR, or SAR module to upload and install.

Path to the new application

● Local file system

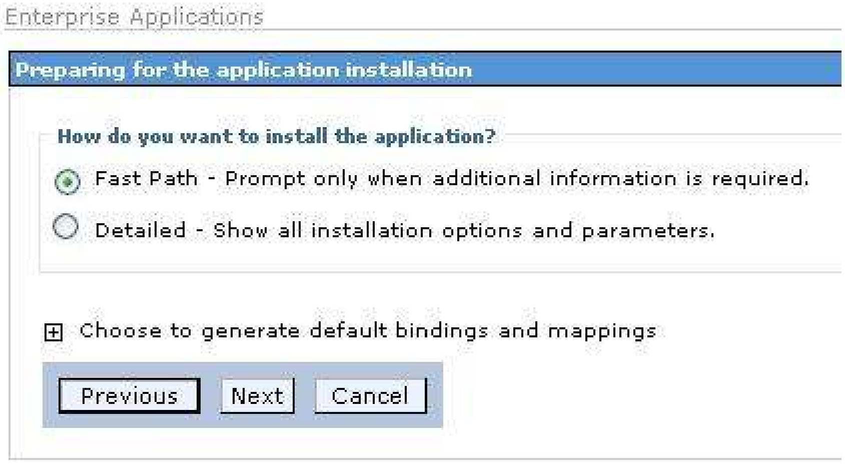
Full path

C:\PT8.50-804-R1\webs

○ Remote file system

Full path

- Select *Fast Path* installation scenario. Click *Next*



Enterprise Applications

Preparing for the application installation

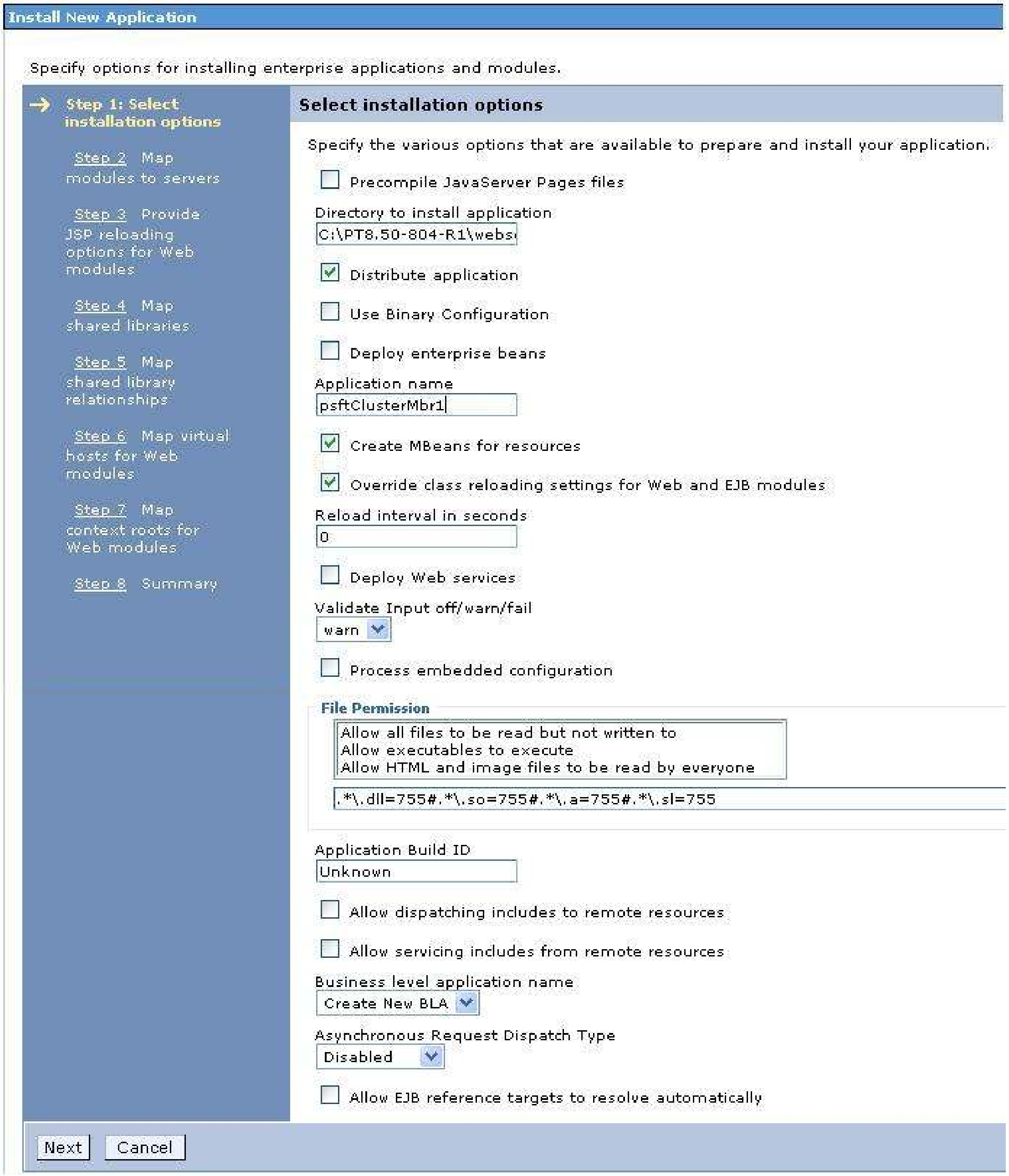
How do you want to install the application?

● Fast Path - Prompt only when additional information is required.

○ Detailed - Show all installation options and parameters.

⊞ Choose to generate default bindings and mappings

- Using the values recorded in step 6b during the PIA install, set *Directory to Install Application* to *PS_HOME\webserv\<appname>\installedApps\<hostname>Node Cell*\ which was the directory that the PIA was installed to on the staging machine or *WebSphere 1* instance. This value must remain the same because the PIA configuration files/scripts contains hard coded references to directory structures. Change *Application Name* to the value specified in step 6b.


Install New Application
→ Step 1: Select installation options
Step 2 Map modules to servers
Step 3 Provide JSP reloading options for Web modules
Step 4 Map shared libraries
Step 5 Map shared library relationships
Step 6 Map virtual hosts for Web modules
Step 7 Map context roots for Web modules
Step 8 Summary
Next
Cancel

Specify options for installing enterprise applications and modules.

Select installation options

Specify the various options that are available to prepare and install your application.

☐ Precompile JavaServer Pages files

Directory to install application: C:\PT8.50-804-R1\webs

☑ Distribute application

☐ Use Binary Configuration

☐ Deploy enterprise beans

Application name: psftClusterMbr1

☑ Create MBeans for resources

☑ Override class reloading settings for Web and EJB modules

Reload interval in seconds: 0

☐ Deploy Web services

Validate Input off/warn/fail: warn

☐ Process embedded configuration

File Permission

Allow all files to be read but not written to
Allow executables to execute
Allow HTML and image files to be read by everyone

.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755

Application Build ID: Unknown

☐ Allow dispatching includes to remote resources

☐ Allow servicing includes from remote resources

Business level application name: Create New BLA

Asynchronous Request Dispatch Type: Disabled

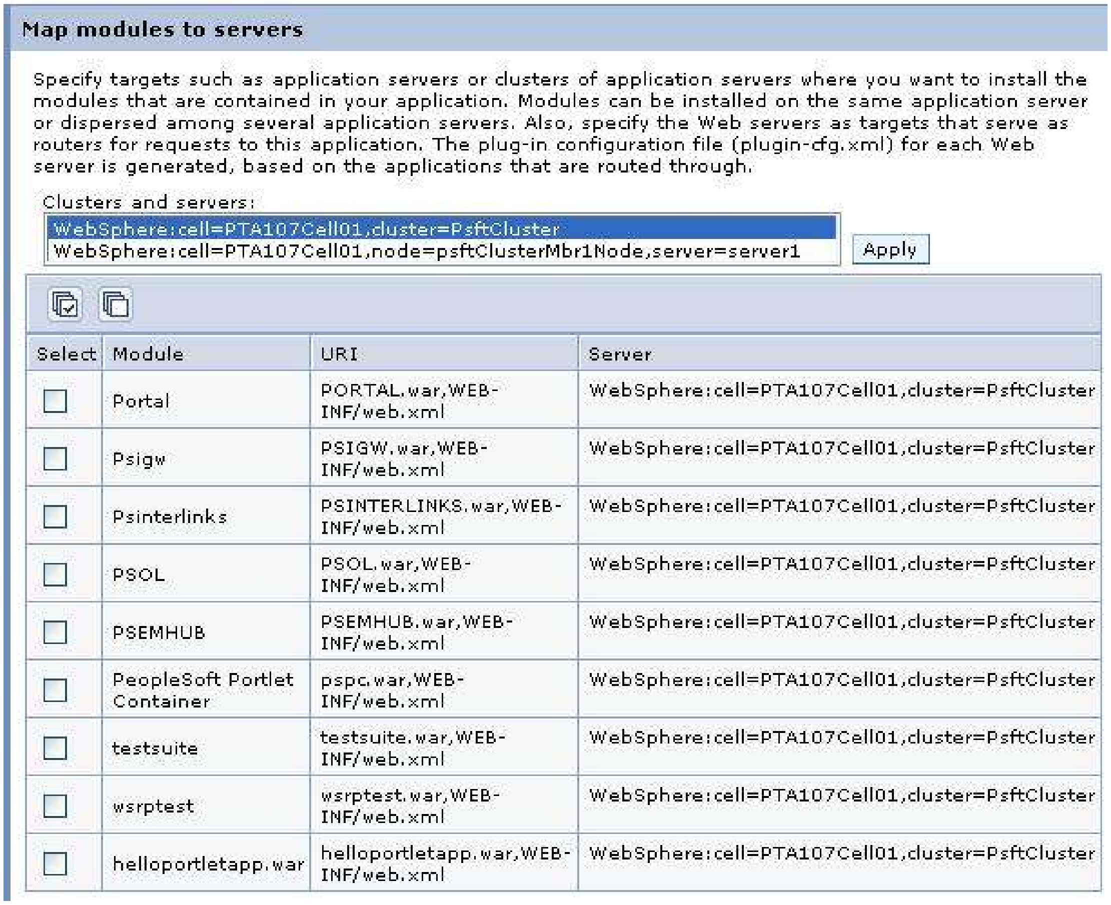
☐ Allow EJB reference targets to resolve automatically

- **Note**: If you want to enter any other *PS_HOME* and *Application Name*, then beware of making the following changes inorder for the application to work,

  ✓ Open *web.xml* file on respective cluster members at the location *PS_HOME/webserv/<appname>/installedApps/<hostname> NodeCell/< application_name>/PORTAL* and change *<param-value>* in *<init-param>* to appropriate *PS_HOME*. Similarly change can be made to web.xml for *PSIGW* Web Module

- ✓ Make sure to correct the library paths in *psSharedLibrary* in *Environment -> Shared Libraries* screen for each cluster member.
- ✓ Make sure to correct the path values for JVM custom properties *java.util.logging.config.file* and *ps_vault* for each cluster member.
- ✓ Inorder for certain utility scripts in the folder *PS_HOME/webserv/<appname>/installedApps/<hostname> NodeCell/< application_name>/* to work properly make sure to correct the path values in utility scripts like *pskeymanager.cmd (or .sh), HashKeyGenerator.bat (or .sh), StartSendMaster.bat (or .sh)* etc

- In *Map modules to servers* screen select all the listed web modules and select the *PsftCluster* as the target cluster for entire application. Click *Apply*. The server mappings should change to match the cluster name.



Map modules to servers

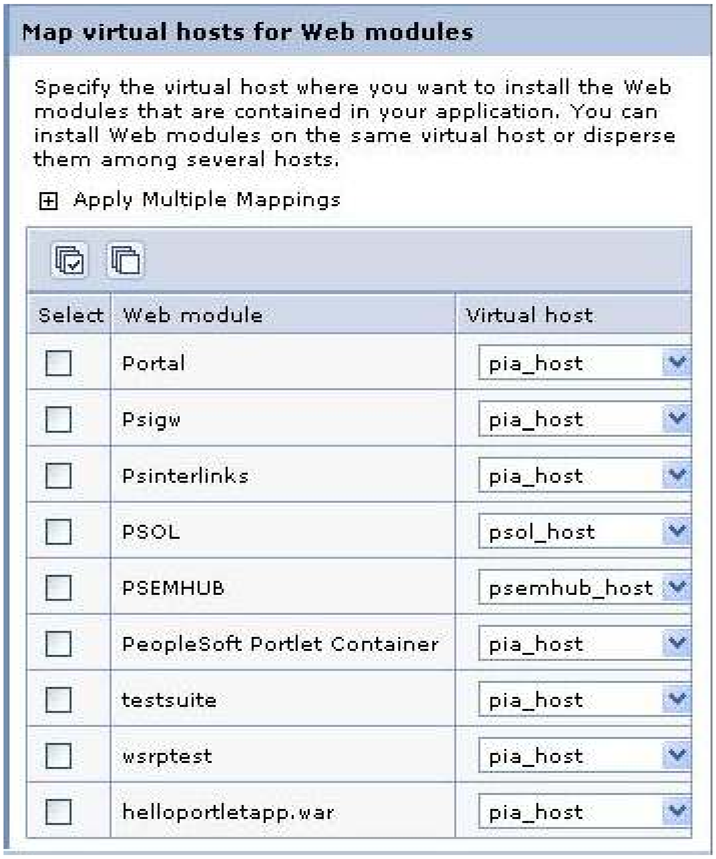
Specify targets such as application servers or clusters of application servers where you want to install the modules that are contained in your application. Modules can be installed on the same application server or dispersed among several application servers. Also, specify the Web servers as targets that serve as routers for requests to this application. The plug-in configuration file (plugin-cfg.xml) for each Web server is generated, based on the applications that are routed through.

Clusters and servers:

WebSphere:cell=PTA107Cell01,cluster=PsftCluster
WebSphere:cell=PTA107Cell01,node=psftClusterMbr1Node,server=server1

Select	Module	URI	Server
☐	Portal	PORTAL.war,WEB-INF/web.xml	WebSphere:cell=PTA107Cell01,cluster=PsftCluster
☐	Psigw	PSIGW.war,WEB-INF/web.xml	WebSphere:cell=PTA107Cell01,cluster=PsftCluster
☐	Psinterlinks	PSINTERLINKS.war,WEB-INF/web.xml	WebSphere:cell=PTA107Cell01,cluster=PsftCluster
☐	PSOL	PSOL.war,WEB-INF/web.xml	WebSphere:cell=PTA107Cell01,cluster=PsftCluster
☐	PSEMHUB	PSEMHUB.war,WEB-INF/web.xml	WebSphere:cell=PTA107Cell01,cluster=PsftCluster
☐	PeopleSoft Portlet Container	pspc.war,WEB-INF/web.xml	WebSphere:cell=PTA107Cell01,cluster=PsftCluster
☐	testsuite	testsuite.war,WEB-INF/web.xml	WebSphere:cell=PTA107Cell01,cluster=PsftCluster
☐	wsrptest	wsrptest.war,WEB-INF/web.xml	WebSphere:cell=PTA107Cell01,cluster=PsftCluster
☐	helloportletapp.war	helloportletapp.war,WEB-INF/web.xml	WebSphere:cell=PTA107Cell01,cluster=PsftCluster

- In *Map virtual hosts for Web modules* screen, select appropriate virtual hosts for each web module. *PSOL* module should be assigned to *psol_host*, *PSEMHUB* module should be assigned to *psemhub_host* and *PORTAL* and other web modules should be assigned to *pia_host*.



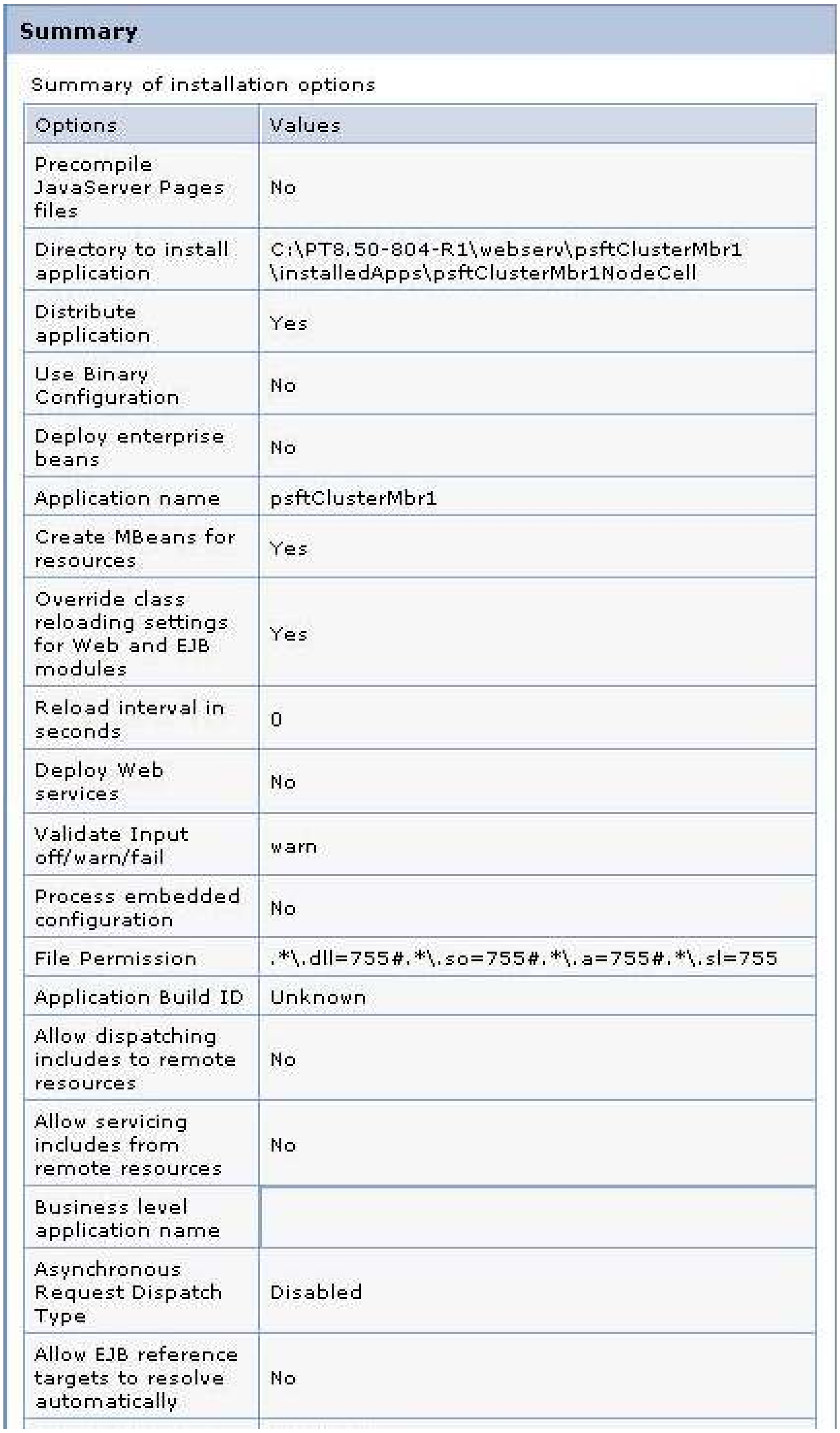
Map virtual hosts for Web modules

Specify the virtual host where you want to install the Web modules that are contained in your application. You can install Web modules on the same virtual host or disperse them among several hosts.

⊞ Apply Multiple Mappings

Select	Web module	Virtual host
□	Portal	pia_host
□	Psigw	pia_host
□	Psinterlinks	pia_host
□	PSOL	psol_host
□	PSEMHUB	psemhub_host
□	PeopleSoft Portlet Container	pia_host
□	testsuite	pia_host
□	wsrptest	pia_host
□	helloportletapp.war	pia_host

- On the *Summary* screen, review your choices. If a problem is seen, click *Previous* to correct the error. Make sure that the *Application Name* and the *Directory to Install Application* match the values used in the PIA installation in step 6b. Click *Finish*. This step will take a few minutes. If a cluster member is on a different *PS_HOME* then you need to manually deploy the EAR file using *EARExpander -operation expand -expansionFlags war* option.



Summary

Summary of installation options

Options	Values
Precompile JavaServer Pages files	No
Directory to install application	C:\PT8.50-804-R1\webserv\psftClusterMbr1 \installedApps\psftClusterMbr1NodeCell
Distribute application	Yes
Use Binary Configuration	No
Deploy enterprise beans	No
Application name	psftClusterMbr1
Create MBeans for resources	Yes
Override class reloading settings for Web and EJB modules	Yes
Reload interval in seconds	0
Deploy Web services	No
Validate Input off/warn/fail	warn
Process embedded configuration	No
File Permission	.*\.dll=755#.*\.so=755#.*\.a=755#.*\.sl=755
Application Build ID	Unknown
Allow dispatching includes to remote resources	No
Allow servicing includes from remote resources	No
Business level application name	
Asynchronous Request Dispatch Type	Disabled
Allow EJB reference targets to resolve automatically	No

- Select option *Save to Master Configuration*. You will see a page that displays the status of the PeopleSoft application install as successful. Check the option *Synchronize changes with Nodes*. Click *Save* button. This step will take a few minutes to copy the application to all the nodes. The Configuration for all servers will be updated, and the EAR file will be transferred and unpacked on all attached servers

- **Renaming JSESSIONID**

- ✓ Go to *Enterprise Applications -> PeopleSoft* application and select *Manage Modules* link
- ✓ Click on *Portal.war*
- ✓ Click on *Session Management* under *Additional properties*. Click *Enable cookies* link and set the *Cookie Name* to be of the format as *<hostname>-<PIAport>-PORTAL-PSJSESSIONID* (for example *PTA107-10001-PORTAL-PSJSESSIONID*)
- ✓ If you have entered *AuthtokenDomain* during the PIA installation enter the same for *Cookie domain* (for example *.peoplesoft.com*) and click *Apply*
- ✓ Click on *Session Management* under *Additional properties* and make sure to select the check box for *Override session management* (ie., enable this option)
- ✓ Click on *PeopleSoft Portlet Container (PSPC.war)* module from the table (when you click on *Manage Modules*)
- ✓ Follow the same steps for *PSPC.war* module and set the *Cookie Name* to be of the format as *<hostname>-<PIAport>-PORTLET-PSJSESSIONID* (for example *PTA107-10001-PORTLET -PSJSESSIONID*)
- ✓ If you have entered *AuthtokenDomain* during the PIA installation enter the same for *Cookie domain* (for example *.peoplesoft.com* and Click *Apply*

- Make sure to save all changes made to the configuration by selecting *Save changes to master repository* from under *System Administration* menu. Before saving make sure any change made to configuration gets propagated to all cluster nodes by enabling the option *Synchronize changes with Nodes* in *Console Preferences* screen.

15. If you have configured *IBM HTTP Server* (or any other webserver like *Sun Java or IIS*) as Reverse Proxy in front of the Deployment Manager, make sure to add the HTTP and HTTPS port entries of the RPS server in virtual hosts *pia_host, psol_host and psemhub_host*. Then generate the webserver plugin *plugin-cfg.xml* file and propagate it to the location where you have created the RPS webserver definition.

16. Start the Cluster

- Go to *Servers -> Clusters -> Application server clusters* screen
- Select *PsftCluster* cluster. Click *Start*

- When all of the servers in the cluster have started the status icon will update to a solid green arrow. This usually takes a few minutes.

- If a message appears warning that a cluster member will not be started, check that the node agent is running on that machine (use the *startNode.bat (or .sh)* command located in *<PS_HOME>/webserv/<appname>/bin*). Once that node agent is started, select the cluster and click start to start the cluster member

Additional Resources

➢ **WebSphere Network Deployment V7.0 Info Center**

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.multiplatform.doc/info/welcome_nd.html

➢ **WebSphere V7.0 Supported Platforms**

http://www-01.ibm.com/support/docview.wss?rs=180&uid=swg27012284

➢ **WebSphere Redbooks** (http://www.redbooks.ibm.com)

WebSphere Application Server V7: Concepts, Planning and Design (SG24-7708-00)

WebSphere Application Server V7.0: Technical Overview (REDP-4482-00)

Books from WebSphere 7.0 ND Library

CONFIGURING AN ORACLE APPLICATION SERVER CLUSTER WITH PEOPLETOOLS 8.47 – 8.48

Oracle Application Server Clustering Process Overview

To create an OAS cluster, the PIA must first be deployed to an OC4J component of OAS Instance and then customized (adding additional PeopleSoft sites, html customizations, etc.). The PIA deployed to the OC4J instance will then be exported as an ear file. The ear file can then be deployed to a cluster. **It is strongly recommended that the initial OC4J instance be used as a staging server to update with patches and customizations to the PIA**. Updates can be exported as an ear file to be deployed to the cluster.

NOTE:

The staging server should be on the same operating system as the production environment. The hard coded directory links in the PIA require that the PIA ear be installed to the same directory in each environment.

There are some extra steps required to deploy PIA to an OAS cluster. Here is a high level overview of the clustering process. The step-by-step process follows in the next section.

1. Install OAS.
2. Install the PIA into a single or multiple OAS OC4J instance(s).
3. Manually recreate the ear file(s).
4. Undeploy the application(s) from the OAS OC4J instance(s)
5. Create a Farm as new file-based repository.
6. Add the OAS instances to the farm.
7. Setup the cluster between the instances.
8. Deploy the manually recreated ear file onto the OAS OC4J instance.
9. Post cluster setup.

Note: Steps 7 & 8 can be interchanged

Recreating application EAR file

Single Component Deployment

- Recreating peoplesoft-OAS.ear file
    1. Create a temp directory, example C:\oracleEAR
    2. Copy the $ORACLE_HOME/j2ee/<application-name>/applications/<application-name> directory into the newly created temp directory C:\oracleEAR
    3. Create WAR files
        a. PORTAL
        cd C:/oracleEAR/<application-name>/PORTAL

```
jar -cvf ../PORTAL.war *
cd ..
```
Remove directory C:/oracleEAR/<application-name>/PORTAL

b. PSIGW
   ```
   cd C:/oracleEAR/<application-name>/PSIGW
   jar -cvf ../PSIGW.war *
   cd ..
   ```
   Remove directory C:/oracleEAR/<application-name>/PSIGW

c. PSINTERLINKS
   ```
   cd C:/oracleEAR/<application-name>/PSINTERLINKS
   jar -cvf ../PSINTERLINKS.war *
   cd ..
   ```
   Remove directory C:/oracleEAR/<application-name>/PSINTERLINKS

d. PSEMHUB
   ```
   cd C:/oracleEAR/<application-name>/PSEMHUB
   jar -cvf ../PSEMHUB.war *
   cd ..
   ```
   Remove directory C:/oracleEAR/<application-name>/PSEMHUB

e. PSOL
   ```
   cd C:/oracleEAR/<application-name>/PSOL
   jar -cvf ../PSOL.war *
   cd ..
   ```
   Remove directory C:/oracleEAR/<application-name>/PSOL

f. pspc
   ```
   cd C:/oracleEAR/<application-name>/pspc
   jar -cvf ../pspc.war *
   cd ..
   ```
   Remove directory C:/oracleEAR/<application-name>/pspc

g. helloportletapp
   ```
   cd C:/oracleEAR/<application-name>/helloportletapp
   jar -cvf ../helloportletapp.war *
   cd ..
   ```
   Remove directory C:/oracleEAR/<application-name>/helloportletapp

h. wsrptest
   ```
   cd C:/oracleEAR/<application-name>/wsrptest
   jar -cvf ../wsrptest.war *
   cd ..
   ```
   Remove directory C:/oracleEAR/<application-name>/wsrptest

i. testsuite
cd C:/oracleEAR/<application-name>/testsuite
*jar -cvf ../testsuite.war **
cd ..
Remove directory C:/oracleEAR/<application-name>/testsuite

4. Finally create the peoplesoft-OAS.ear file
cd C:/oracleEAR/<application-name>
*jar -cvf ../peoplesoft-OAS.ear **

Multi Component Deployment

- Recreating PIA.ear file

1. Create a temp directory, example C:\oracleEAR
2. Copy the $ORACLE_HOME/j2ee/<PIA_application-name>/applications/<application-name> directory into the newly created temp directory C:\oracleEAR
3. Create WAR files

a. PORTAL
cd C:/oracleEAR/<application-name>/PORTAL
*jar -cvf ../PORTAL.war **
cd ..
Remove directory C:/oracleEAR/<application-name>/PORTAL

b. PSIGW
cd C:/oracleEAR/<application-name>/PSIGW
*jar -cvf ../PSIGW.war **
cd ..
Remove directory C:/oracleEAR/<application-name>/PSIGW

c. PSINTERLINKS
cd C:/oracleEAR/<application-name>/PSINTERLINKS
*jar -cvf ../PSINTERLINKS.war **
cd ..
Remove directory C:/oracleEAR/<application-name>/PSINTERLINKS

d. pspc
cd C:/oracleEAR/<application-name>/pspc
*jar -cvf ../pspc.war **
cd ..
Remove directory C:/oracleEAR/<application-name>/pspc

e. helloportletapp
cd C:/oracleEAR/<application-name>/helloportletapp
*jar -cvf ../helloportletapp.war **
cd ..
Remove directory C:/oracleEAR/<application-name>/helloportletapp

f. wsrptest
cd C:/oracleEAR/<application-name>/wsrptest
*jar -cvf ../wsrptest.war **
cd ..
Remove directory C:/oracleEAR/<application-name>/wsrptest

g. testsuite
cd C:/oracleEAR/<application-name>/testsuite
*jar -cvf ../testsuite.war **
cd ..
Remove directory C:/oracleEAR/<application-name>/testsuite

4. Finally create the PIA.ear file
cd C:/oracleEAR/<application-name>
*jar -cvf ../PIA.ear **

- Recreating PSEMHUB.ear file

1. Create a temp directory, example C:\oracleEAR

2. Copy the $ORACLE_HOME/j2ee/<PSEMHUB_application-name>/applications/<application-name> directory into the newly created temp directory C:\oracleEAR

3. Create WAR files

   a. PSEMHUB
   cd C:/oracleEAR/<application-name>/PSEMHUB
   *jar -cvf ../PSEMHUB.war **
   cd ..
   Remove directory C:/oracleEAR/<application-name>/PSEMHUB

4. Finally create the PSEMHUB.ear file
cd C:/oracleEAR/<application-name>
*jar -cvf ../PSEMHUB.ear **

- Recreating PSOL.ear file

1. Create a temp directory, example C:\oracleEAR

2. Copy the $ORACLE_HOME/j2ee/<PSOL_application-name>/applications/<application-name> directory into the newly created temp directory C:\oracleEAR

3. Create WAR files

    a. PSOL
    cd C:/oracleEAR/<application-name>/PSOL
    *jar -cvf ../PSOL.war **
    cd ..
    Remove directory C:/oracleEAR/<application-name>/PSOL

4. Finally create the PSOL.ear file
    cd C:/oracleEAR/<application-name>
    *jar -cvf ../PSOL.ear **

DCM-Managed Oracle Application Server Clusters

When administering a DCM-Managed OracleAS Cluster, an administrator uses either Application Server Control Console or dcmctl commands to manage and configure common configuration information on one Oracle Application Server instance. DCM then propagates and replicates the common configuration information across all Oracle Application Server instances within the DCM-Managed OracleAS Cluster. The common configuration information for the cluster is called the cluster-wide configuration.

Each application server instance in a DCM-Managed OracleAS Cluster has the same base configuration. The base configuration contains the cluster-wide configuration and excludes instance-specific parameters.

Creating DCM-Managed OracleAS Clusters

An OracleAS Farm contains a collection of Oracle Application Server instances. In an OracleAS Farm, you can view a list of all application server instances when you start Application Server Control Console. The application server instances shown in the Standalone Instances area on the Application Server Control Console Farm Home Page are available to be added to DCM-Managed OracleAS Clusters.

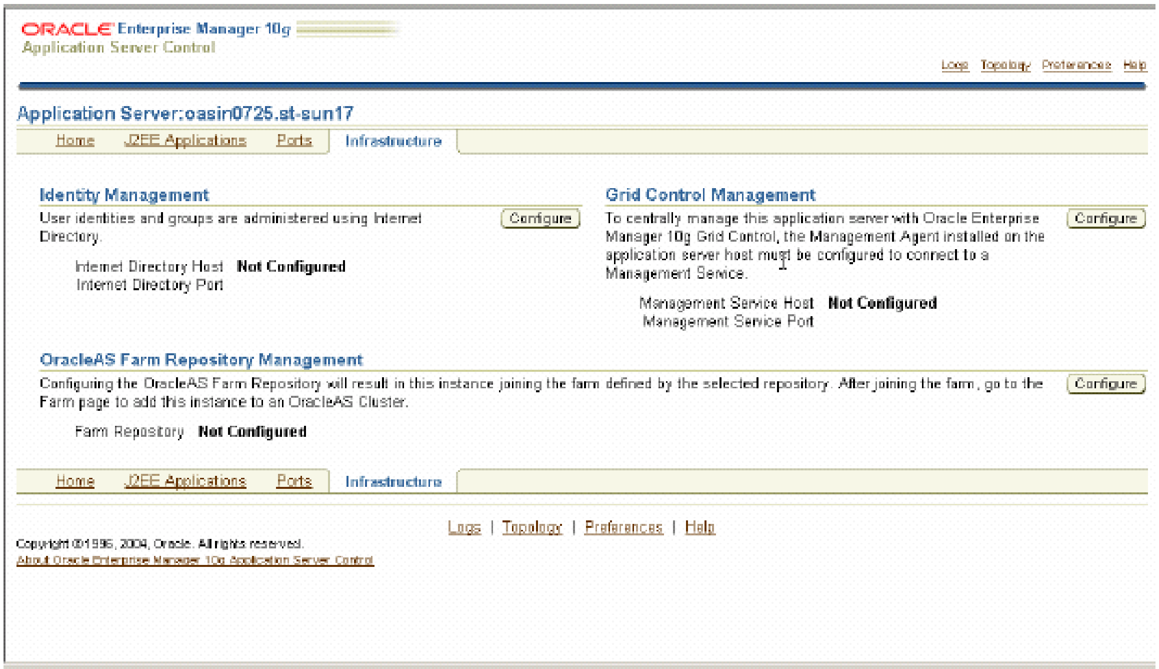
Each Oracle Application Server Farm has the characteristic that it uses either a File Based Repository or a Database-Based Repository. The steps for associating an application server instance with an OracleAS Farm differ depending on the type of the respiratory.

Creating an OracleAS File-based Farm Repository Host

You can instruct the Oracle Application Server installer to create an OracleAS File-based Farm when you install Oracle Application Server. If you did not create an OracleAS Filebased Farm during installation, then you can create the OracleAS File-based Farm with the following steps.

1. Using the Application Server Control Console for the instance that you want to use as the repository host, select the Infrastructure link to navigate to the Infrastructure page. If a repository is not configured, then the Farm Repository field shows "Not Configured", as shown in Figure-1.

Figure -1 Application Server Control Console Farm Repository Management


ORACLE Enterprise Manager 10g
Application Server Control
Logs Topology Preferences Help
Home J2EE Applications Ports Infrastructure
Configure
Configure
Configure
Home J2EE Applications Ports Infrastructure
Logs | Topology | Preferences | Help

Application Server:oasin0725.st-sun17

Identity Management

User identities and groups are administered using Internet Directory.

Internet Directory Host **Not Configured**
Internet Directory Port

Grid Control Management

To centrally manage this application server with Oracle Enterprise Manager 10g Grid Control, the Management Agent installed on the application server host must be configured to connect to a Management Service.

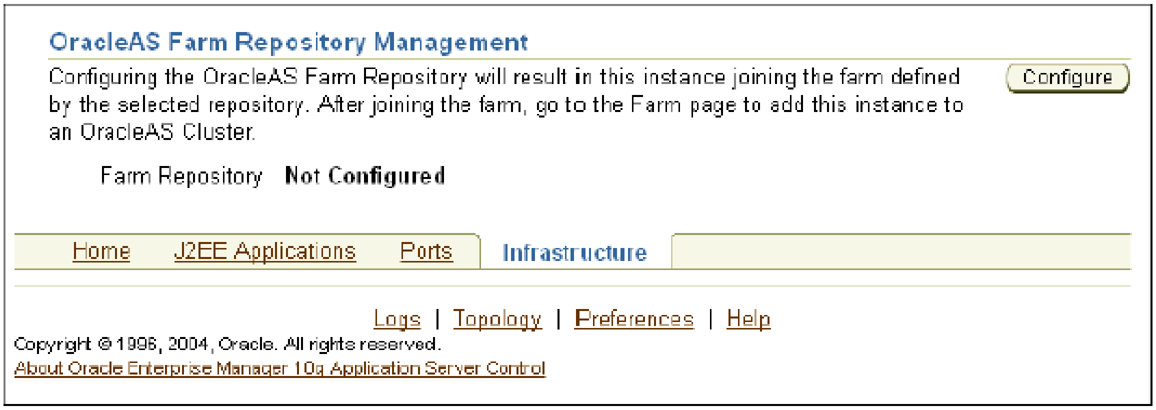
Management Service Host **Not Configured**
Management Service Port

OracleAS Farm Repository Management

Configuring the OracleAS Farm Repository will result in this instance joining the farm defined by the selected repository. After joining the farm, go to the Farm page to add this instance to an OracleAS Cluster.

Farm Repository **Not Configured**

Copyright ©1996, 2004, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Application Server Control


Configure
Home J2EE Applications Ports Infrastructure
Logs | Topology | Preferences | Help

OracleAS Farm Repository Management

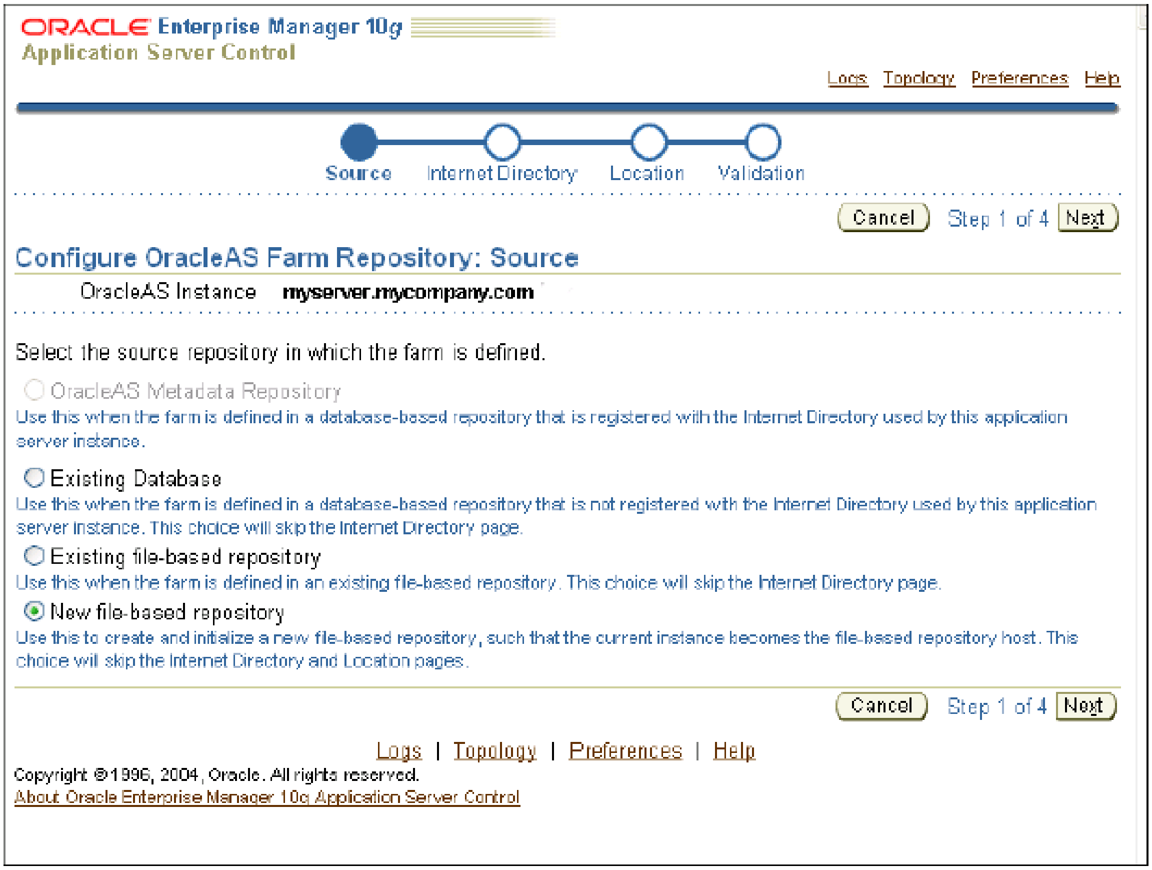
Configuring the OracleAS Farm Repository will result in this instance joining the farm defined by the selected repository. After joining the farm, go to the Farm page to add this instance to an OracleAS Cluster.

Farm Repository **Not Configured**

Copyright © 1996, 2004, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Application Server Control

2. On the Infrastructure page, in the OracleAS Farm Repository Management area, select the **Configure** button to start the Configure OracleAS Farm Repository wizard. The repository creation wizard appears. The appropriate host name appears under Configure Oracle Farm Repository Source. Select the **New file-based repository** button and select **next**, as shown in Figure -2.

Figure -2 Application Server Control Console Create Repository Wizard Step 1


ORACLE Enterprise Manager 10g
Application Server Control
Logs Topology Preferences Help
Source
Internet Directory
Location
Validation
Cancel
Step 1 of 4
Next

Configure OracleAS Farm Repository: Source

OracleAS Instance myserver.mycompany.com

Select the source repository in which the farm is defined.

○ OracleAS Metadata Repository
Use this when the farm is defined in a database-based repository that is registered with the Internet Directory used by this application server instance.

○ Existing Database
Use this when the farm is defined in a database-based repository that is not registered with the Internet Directory used by this application server instance. This choice will skip the Internet Directory page.

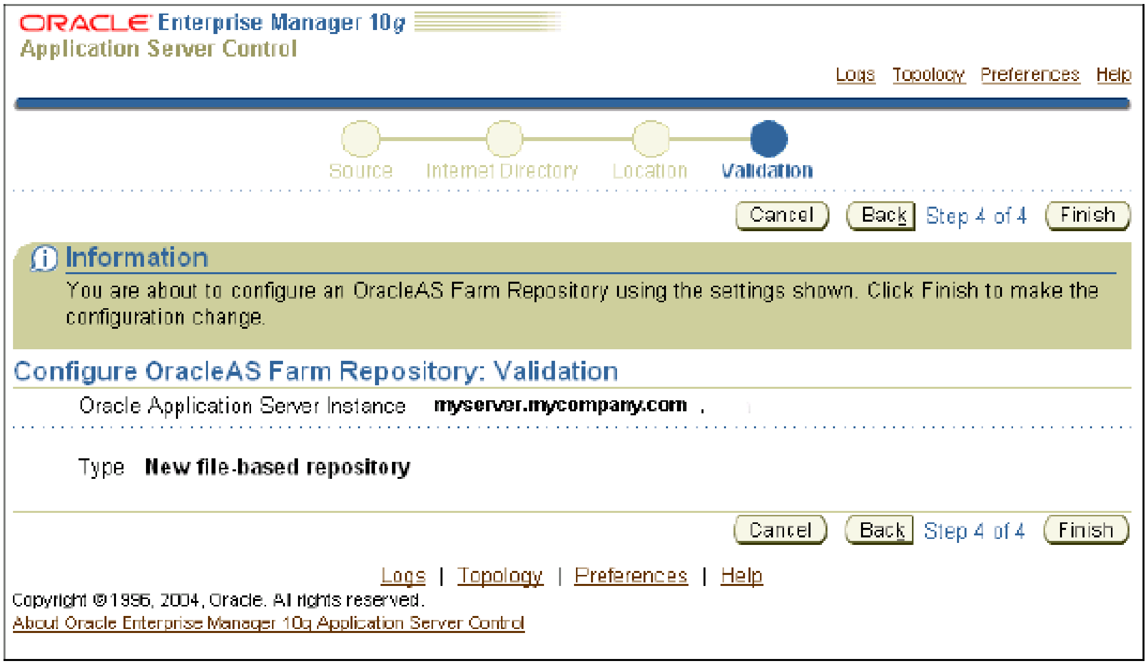
○ Existing file-based repository
Use this when the farm is defined in an existing file-based repository. This choice will skip the Internet Directory page.

● New file-based repository
Use this to create and initialize a new file-based repository, such that the current instance becomes the file-based repository host. This choice will skip the Internet Directory and Location pages.

Copyright ©1996, 2004, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Application Server Control

3. The wizard jumps to Step 4 of 4, Validation, as shown in Figure -3.

Figure -3 Application Server Control Console Create Repository Wizard Step 4


ORACLE Enterprise Manager 10g
Application Server Control
Logs Topology Preferences Help
Source
Internet Directory
Location
Validation
Cancel
Back
Step 4 of 4
Finish

Information

You are about to configure an OracleAS Farm Repository using the settings shown. Click Finish to make the configuration change.

Configure OracleAS Farm Repository: Validation

Oracle Application Server Instance myserver.mycompany.com .

Type New file-based repository

Cancel
Back
Step 4 of 4
Finish
Logs | Topology | Preferences | Help
Copyright © 1996, 2004, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Application Server Control

4. Select **Finish** and Oracle Application Server creates the OracleAS File-based Farm.

5. When the wizard completes, note the Repository ID shown in the OracleAS Farm Repository Management area on the Infrastructure page. You need to use the Repository ID to add instances to the OracleAS File-based Farm.

When you go to the Application Server Control Console Home page, notice that the home page shows the OC4J instance and the Oracle HTTP Server are stopped, and the page now includes a Farm link in the General area.

Adding Instances to an OracleAS File-based Farm

To add standalone application server instances to an OracleAS File-based Farm, perform the following steps:

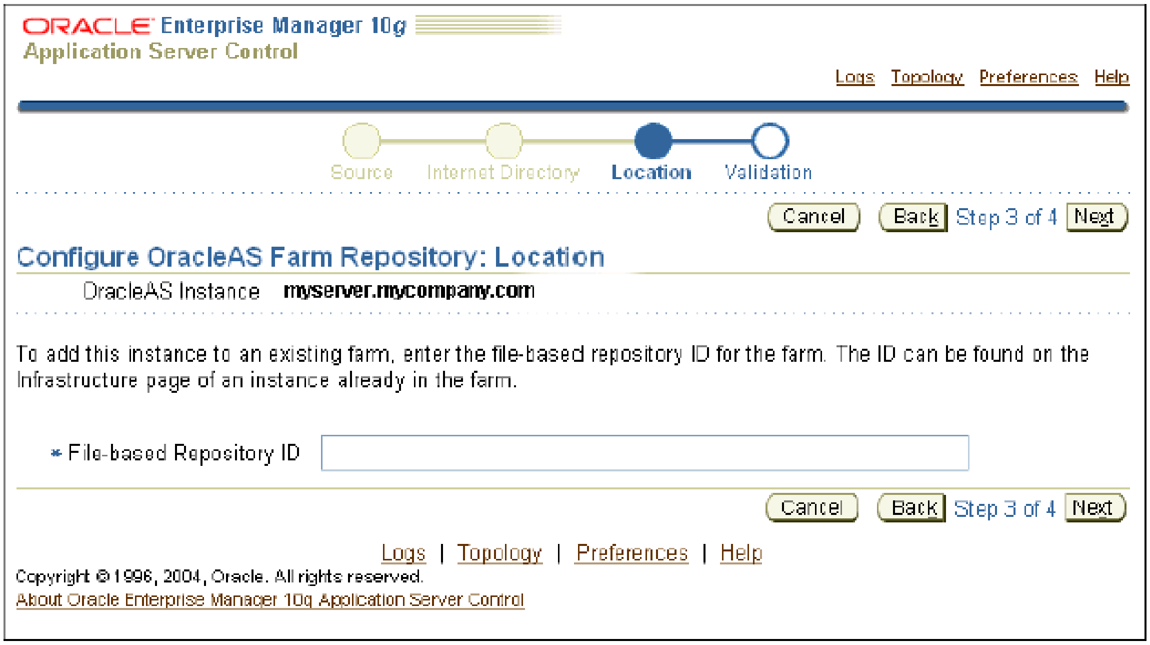
1. Obtain the Repository ID for the OracleAS File-based Farm that you want to join. To find the Repository ID, on any Oracle Application Server instance that uses the OracleAS File-based Farm, select the Infrastructure link, and check the value of the File-based Repository ID field in the OracleAS Farm Repository Management area.

2. Switch to the Application Server Control Console for the standalone instance that you want to add to the OracleAS File-based Farm and select the Infrastructure link. If a repository is not configured, then the Farm Repository field shows "Not Configured", as shown in Figure -1.

3. Select the **Configure** button to start the Configure OracleAS Farm Repository wizard. The repository creation wizard appears, as shown in Figure -2. The appropriate host name appears in the OracleAS Instance field under the Configure Oracle Farm Repository Source area.

4. Select the **Existing file-based repository** button and select **Next**. The repository reation wizard then brings up the Location page, Step 3 of 4, as shown in Figure -4.

Figure -4 Application Server Control Console Add Instance to Farm



ORACLE Enterprise Manager 10g
Application Server Control
Logs Topology Preferences Help
Source
Internet Directory
Location
Validation
Cancel
Back
Step 3 of 4
Next

Configure OracleAS Farm Repository: Location

OracleAS Instance myserver.mycompany.com

To add this instance to an existing farm, enter the file-based repository ID for the farm. The ID can be found on the Infrastructure page of an instance already in the farm.

* File-based Repository ID

Cancel
Back
Step 3 of 4
Next
Logs | Topology | Preferences | Help

Copyright © 1996, 2004, Oracle. All rights reserved.

About Oracle Enterprise Manager 10g Application Server Control

5. Enter the repository ID for the Repository Host and select **Next**.

6. This shows wizard Step 4 of 4 page, Configure OracleAS Farm Repository Validation. Select **Finish**. When the wizard completes, the standalone instance joins the OracleAS File-based Farm.

7. After the wizard completes, you return to the Application Server Control Console Infrastructure page.

Using the Application Server Control Console Create Cluster Page

Using the Application Server Control Console Farm Home Page, you can create a new DCM-Managed OracleAS Cluster.

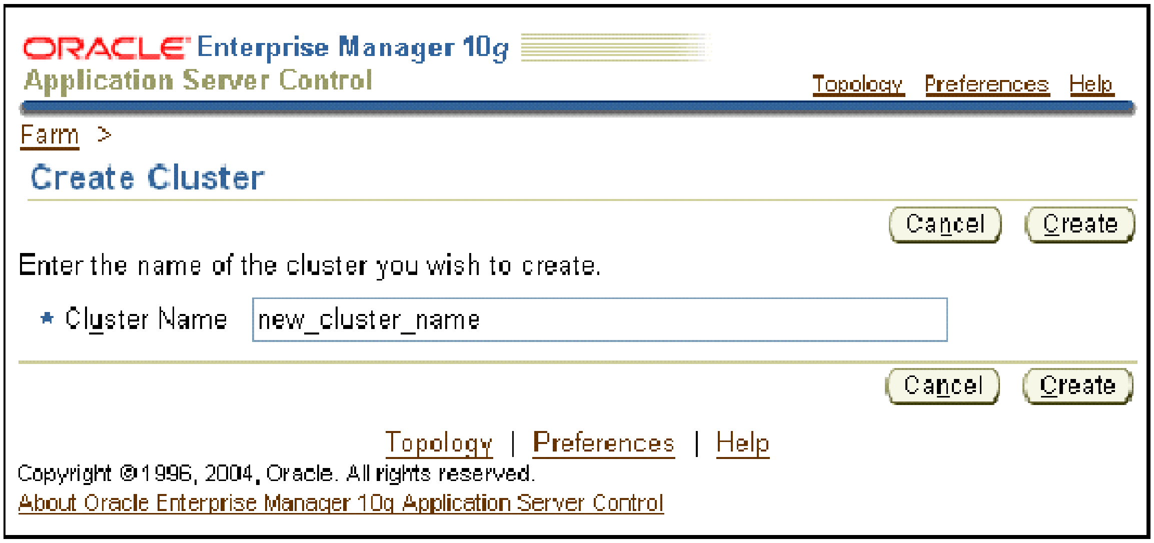
From the Farm Home page, create a new DCM-Managed OracleAS Cluster as follows:

1. Select the Farm link to navigate to the Farm Home Page.

> Note:
>
> Application Server Control Console shows the Farm Home Page when an Oracle Application Server instance is part of a farm

2. Select the **Create Cluster** button. Application Server Control Console displays the Create Cluster page as shown in Figure -5.

Figure -5 Create Cluster Page


ORACLE Enterprise Manager 10g
Application Server Control
Topology Preferences Help
Farm >
Create Cluster
Cancel Create

Enter the name of the cluster you wish to create.

* Cluster Name: new_cluster_name

Cancel Create
Topology | Preferences | Help

Copyright © 1996, 2004, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Application Server Control

3. Enter a name for the new cluster and click **Create.** Each new cluster name within the farm must be unique.

   A confirmation page appears.

4. Click **OK** to return to the Farm Home Page.

After creating a new cluster, the Farm Home page shows the cluster in the Clusters area. After creating a new cluster, the cluster is empty and does not

include any application server instances.
Use the **Join Cluster** button on the Farm Home page to add application server instances to a Cluster.

Adding Instances To DCM-Managed OracleAS Clusters

To add application server instances to a DCM-Managed OracleAS Cluster, do the following:

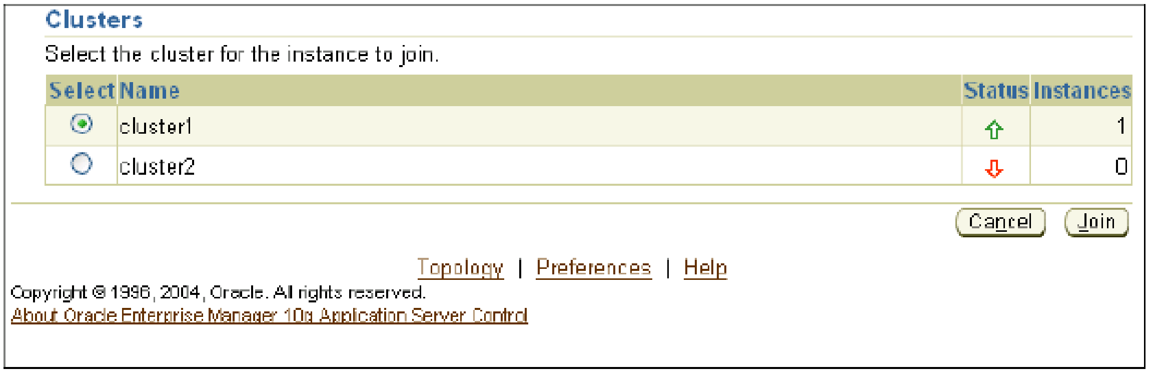
1. Navigate to the Farm Home Page. To navigate to the Farm Home page from an Oracle Application Server instance Home page, select the link next to the Farm field in the General area on the Home page.

> Note:
>
> If the Farm field is not shown, then the instance is not part of a Farm and you will need to associate the standalone instance with a Farm.

2. Select the radio button for the application server instance that you want to add to a cluster from the Standalone Instances section.
3. Click **Join Cluster**.

   Figure -6 shows the Join Cluster page.

   Figure -6 Join Cluster Page



Clusters

Select the cluster for the instance to join.

Select	Name	Status	Instances
●	cluster1	⇧	1
○	cluster2	⇩	0

Copyright © 1996, 2004, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Application Server Control

4. Select the radio button of the cluster that you want the application server instance to join.
5. Click **Join**. OracleAS adds the application server instance to the selected cluster and then displays a confirmation page.

6. Click **OK** to return to the Farm Home Page.

Repeat these steps for each additional standalone application server instance you want to join he cluster.

Note the following when adding application server instances to a DCM-Managed OracleAS Cluster:

1. When adding application server instances to a DCM-Managed OracleAS Cluster, the order that you add instances is significant. The first application server instance that joins the DCM-Managed OracleAS Cluster is used as the base configuration for all additional application server instances that join the cluster. The base configuration includes all cluster-wide configuration information. It does not include instance specific parameters.
2. After the first application server instance joins the DCM-Managed OracleAS Cluster, the base configuration overwrites existing cluster-wide configuration information for subsequent application server instances that join the cluster. Each additional application server instance, after the first, that joins the cluster inherits the base configuration specified for the first application server instance that joins the cluster.
3. Before an application server instance joins a DCM-Managed OracleAS Cluster, application Server Control Console stops the instance. You can restart the application server instance by selecting the cluster link, selecting the appropriate instance from within the cluster, and then selecting the **Start** button.
4. An application server instance is removed from the Standalone Instances area when the instance joins a DCM-Managed OracleAS Cluster.
5. To add multiple standalone application server instances to a DCM-Managed OracleAS Cluster in a single operation, use the dcmctl joinCluster command.
6. When an application server instance contains certain Oracle Application Server components, it is not clusterable. Use the dcmctl isClusterable command to test if an application server instance is clusterable. If the application server instance is not clusterable, then Application Server Control Console returns an error when you attempt to add the instance to a DCM-Managed OracleAS Cluster.
7. To be clusterable, all application server instances that are to be members of a DCM-Managed OracleAS Cluster must be installed on the same flavor operating system. For example, different variants of

UNIX are clusterable together, but they are not clusterable with Windows systems.

> Note:
>
> For adding instances to a OracleAS File-based Farm, where the instances will be added to an DCM-Managed OracleAS Cluster, there is no known fixed upper limit on the number of instances; a DCM-Managed OracleAS Cluster of 12 instances has been tested successfully.

Removing Instances from DCM-Managed OracleAS Clusters

To remove an application server instance from a cluster, do the following:

1. Select the cluster in which you are interested on the Farm Home Page. This brings you to the cluster page.
2. Select the radio button of the application server instance to remove from the cluster and click **Remove**.

To remove multiple standalone application server instances, you need to repeat these steps multiple times.

Note the following when removing application server instances from a DCM-Managed OracleAS Cluster:

- Before an application server instance leaves a cluster, Application Server Control Console stops the instance. After the operation completes, you restart the application server instance from the Standalone Instances area of the Farm Home Page.
- The dcmctl leaveCluster command removes one application server instance from the cluster at each invocation.
- When the last application server instance leaves a cluster, cluster-wide configuration information associated with the cluster is removed. The cluster is now empty and the base configuration is not set. Subsequently, Oracle Application Server uses the first application server instance that joins the cluster as the base configuration for all additional application server instances that join the cluster.
- You can remove an application server instance from the cluster at any time. The first instance to join a cluster does not have special properties. The base configuration is created from the first instance

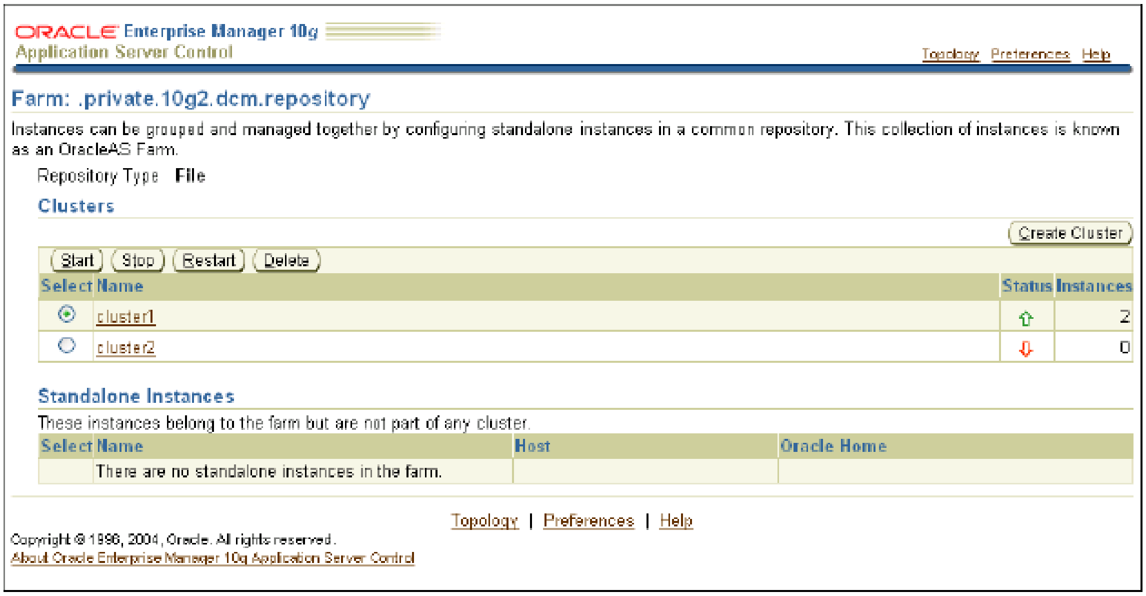
to join the cluster, but this instance can be removed from the cluster in the same manner as the other instances.

- Use the following command to remove the instance form the farm dcmctl leaveFarm

Starting Stopping and Deleting DCM-Managed OracleAS Clusters

Figure -7 shows the Application Server Control Console Farm Home Page, including two clusters, cluster1 and cluster2.

Figure -7 Oracle Application Server 10g Farm Page


ORACLE Enterprise Manager 10g
Application Server Control
Topology Preferences Help
Create Cluster
Start Stop Restart Delete
Topology | Preferences | Help

Farm: .private.10g2.dcm.repository

Instances can be grouped and managed together by configuring standalone instances in a common repository. This collection of instances is known as an OracleAS Farm.

Repository Type File

Clusters

Select	Name	Status	Instances
●	cluster1	⇧	2
○	cluster2	⇩	0

Standalone Instances

These instances belong to the farm but are not part of any cluster.

Select	Name	Host	Oracle Home
	There are no standalone instances in the farm.		

Copyright © 1996, 2004, Oracle. All rights reserved.
About Oracle Enterprise Manager 10g Application Server Control

Table -1 lists the cluster control options available on the Farm Home Page.

Table -1 Oracle Application Server Farm Page Options

If you want to...	Then...
Start all application server instances in a DCM-Managed OracleAS Cluster.	Select the radio button next to the cluster and click **Start.**

Restart all application server instances in a DCM-Managed OracleAS Cluster.	Select the radio button next to the cluster and click **Restart.**
Stop all application server instances in a DCM-Managed.OracleAS Cluster.	Select the radio button next to the cluster and click **Stop**
Delete a DCM-Managed OracleAS Cluster, including any application server instances still included in the cluster (the instances are removed from the cluster and become standalone instances in the Farm).	Select the radio button next to the cluster and click **Delete.**

Using and configuring mod_oc4j Load Balancing

Using DCM-Managed OracleAS Clusters the Oracle HTTP Server module mod_oc4j load balances requests to OC4J processes. The Oracle HTTP Server, using mod_oc4j configuration options, supports different load balancing policies. By specifying load-balancing policies DCM-Managed OracleAS Clusters provide performance benefits along with failover and high vailability, depending on the network topology and host machine capabilities.

By default, mod_oc4j uses weights to select a node to forward a request to. Each node uses a default weight of 1. A node's weight is taken as a ratio compared to the weights of the other available nodes to define the number of requests the node should service compared to the other nodes in the DCM-Managed OracleAS Cluster. Once a node is selected to service a particular request, by default, mod_oc4j uses the roundrobin policy to select OC4J processes on the node. If an incoming request belongs to an established session, the request is forwarded to the same node and the same OC4J process that started the session.

The mod_oc4j load balancing policies do not take into account the number of OC4J processes running on a node when calculating which node to send a request to. Node selection is based on the configured weight for the node, and its availability.

To modify the mod_oc4j load balancing policy, Administrators use the Oc4jSelectMethod and Oc4jRoutingWeight configuration directives in the mod_oc4j.conf file. Using Application Server Control Console, configure the mod_oc4j.conf file as follows:

1. Select the HTTP_Server component from the System Components area of an instance home page.
2. Select the Administration link on the HTTP_Server page.
3. Select the Advanced Server Properties link on the HTTP_Server page Administration page.
4. On the Advanced Server Properties page, select the mod_oc4j.conf link from the Configuration Files area.
5. On the Edit mod_oc4j.conf page, within the <IfModule mod_oc4j.c> section, and specify the directives Oc4jSelectMethod and Oc4jRoutingWeight to select the desired load balancing option.

Note:

If you do not use Application Server Control Console, you can edit mod_oc4j.conf and use the dcmctl updateConfig command to propagate changes to other mod_oc4j.conf files across a DCM-Managed OracleAS Cluster as follows:

```
% dcmctl updateconfig -ct ohs
```

```
% opmnctl @cluster:<cluster_name> restartproc ias-component=HTTP_Server processtype= HTTP_Server
```

Where *cluster_name* is the name of the cluster.

The opmnctl restartproc command is required for the changes to take effect across all the instances in the cluster.

Updating and Checking the State of Local Configuration

It is important that all configuration changes complete successfully, and that all instances in a cluster are "In Sync". The local configuration information must match the information stored in the repository. DCM does not know about manual changes to configuration files, and such changes could make the instances in a cluster have an In Sync status of false.

Use the following dcmctl command to return a list of all managed components with their In Sync status:

```
dcmctl getState -cl cluster_name
```

The In Sync status of true implies that the local configuration information for a component is the same as the information that is stored in the repository.

If you need to update the file-based repository with changed, local information, use the dcmctl command updateConfig, as follows,

```
dcmctl updateconfig
```

```
dcmctl getstate
```

Use the command resyncInstance to update local information with information from the repository. For example,

```
dcmctl resyncinstance
```

By default this command only updates configuration information for components whose In Sync status is false. Use the -force option to update all components, regardless of their In Sync status.

Performing Administration on a DCM-Managed OracleAS Cluster

During planned administrative downtimes, with a DCM-Managed OracleAS Cluster using an OracleAS File-based Farm that runs on multiple hosts with sufficient resources, you can perform ministrative tasks while continuing to handle requests. This section describes how to relocate the repository host in a DCM-Managed OracleAS Cluster, while continuing to handle requests.

These procedures are useful for performing administrative tasks on a DCM-Managed OracleAS Cluster, such as the following:

- Relocating the repository for repository host node decommission.
- Applying required patches to the DCM-Managed OracleAS Cluster.
- Applying system upgrades, changes, or patches that require a system restart for a host in the DCM-Managed OracleAS Cluster.

> Note:
>
> Using the procedures outlined in this section, only administration capabilities are lost during a planned downtime.

Use the following steps to relocate the repository host in a DCM-Managed OracleAS Cluster.

1. Issue the following DCM command,

   On UNIX systems:

   ```
   cd $ORACLE_HOME/dcm/bin
   ```

```
dcmctl exportRepository -f file
```

On Windows systems:

```
cd %ORACLE_HOME%\dcm\bin
dcmctl exportRepository -f file
```

> Note:
>
> After this step, do not perform configuration or administration commands that would change the configuration. Otherwise those changes will not be copied when the repository file is imported to the new repository host.

2. Stop the administrative system, including Enterprise Manager and the DCM daemon in each instance of the OracleAS File-based Farm, except for the instance that is going to be the new repository host.

   On UNIX systems use the following commands on each instance in the cluster:

   ```
   $ORACLE_HOME/bin/emctl stop iasconsole
   $ORACLE_HOME/opmn/bin/opmnctl stopproc ias-component=dcm-daemon
   ```

   On Windows systems use the following commands on each instance in the cluster:

   ```
   %ORACLE_HOME%\bin\emctl stop iasconsole
   %ORACLE_HOME%\opmn\bin\opmnctl stopproc ias-component=dcm-daemon
   ```

   At this point, the DCM-Managed OracleAS Cluster can still handle requests.

3. Import the saved repository on the host that is to be the repository host instance.

   On UNIX systems, use the following commands:

   ```
   cd $ORACLE_HOME/dcm/bin/
   dcmctl importRepository -file file name
   ```

   On Windows systems, use the following commands:

   ```
   cd %ORACLE_HOME%\dcm\bin
   dcmctl importRepository -file file name
   ```

where *file name* is the name of the file you specified in the exportRepository command.
While importRepository is active, the DCM-Managed OracleAS Cluster can still handle requests.

> Note:
>
> The importRepository command issues a prompt that specifies that the system that is the currently hosting the repository must be shutdown. However, only the dcm-daemon on the system that is currently hosting the repository must be shutdown, and not the entire system

4. Use the following command to start all components on the new repository host. Do not erform administrative functions at this time.

   On UNIX systems:

   ```
   $ORACLE_HOME/opmn/bin/opmnctl startall
   ```

   On Windows systems:

   ```
   %ORACLE_HOME%\opmn\bin\opmnctl startall
   ```

5. On the system that was the repository host, indicate that the instance is no longer the host by issuing the following command,
   dcmctl repositoryRelocated

6. Start the Application Server Control Console on the new repository host instance. The repository has now been relocated, and the new repository instance now handles requests.

   On UNIX systems use the following commands on each instance in the cluster:

   ```
   $ORACLE_HOME/bin/emctl start iasconsole
   ```

   On Windows systems use the following commands on each instance in the cluster:

   ```
   %ORACLE_HOME%\bin\emctl start iasconsole
   ```

7. Shut down the Oracle Application Server instance associated with the old repository host, using the following commands:

   On UNIX systems:

   ```
   $ORACLE_HOME/opmn/bin/opmnctl stopall
   ```

   On Windows systems:

```
        %ORACLE_HOME%\opmn\bin\opmnctl startall
```

You can now perform the required administrative tasks on the old repository host system,such as the following.

- Applying required patches to the repository host system in the DCM-Managed OracleAS Cluster.
- Decommission the node.
- Applying system upgrades, changes, or patches that require a system restart for the DCM-Managed OracleAS Cluster.

Best Practices for Managing Instances In OracleAS File-based Farms

When joining or leaving an OracleAS File-based Farm, all the managed processes are shutdown on the instance that is joining or leaving the OracleAS File-based Farm. If you want the instance to be available, restart the instance after performing the leave or join farm operation.

It is recommended that you make a backup of the local configuration before either leaving or joining an OracleAS File-based Farm. For example, use the following command to create an archive:

```
        dcmctl createarchive -arch myarchive -comment "Archive before
leaving xyz farm"

        dcmctl exportarchive -arch myarchive -f /archives/myarchive
```

Archives are portable across OracleAS File-based Farm. When an instance joins a new farm it can apply archives created on a previous farm.

Post Cluster Setup.

Once the cluster is setup, following steps needs to be performed for the cluster to work properly.

1. On the OAS instances that are joining the cluster, copy all jar files from the PIA web application lib directory into the applib directory of the component where PIA web application is deployed.

   For example, if the OC4J component is PeopleSoft and the application name is PeopleSoft, copy all the jar files from $ORACLE_HOME/j2ee/PeopleSoft/applications/PeopleSoft/lib directory into $ORACLE_HOME/j2ee/PeopleSoft/applib directory.

2. PSIGW Web application property file integrationGateway.properties contains some properties that have absolute paths. IntegrationGateway.properties is located in

$ORACLE_HOME/j2ee/<component_name>/applications/<application_name>/PSIGW/WEB-INF directory. Make sure that this file is updated to point to the proper locations for the new OAS instance that is joining the cluster.

CONFIGURING AN ORACLE APPLICATION SERVER CLUSTER WITH PEOPLETOOLS 8.49

In OAS 10.1.3.1 release, a cluster topology is defined as two or more loosely connected Oracle Application Server nodes.

The connectivity provided within a cluster is a function of Oracle Notification Server (ONS), which manages communications between Oracle Application Server components, including OC4J and Oracle HTTP Server. The ONS server is a component of Oracle Process Manager and Notification Server (OPMN), which is installed by default on every Oracle Application Server host. When configuring a cluster topology, you are actually connecting the ONS servers running on each Oracle Application Server node.

Changes in Clustering

The following are changes in clustering configuration in Oracle Application Server 10g Release 3 (10.1.3.1) from previous releases.

1. As of Oracle Application Server 10g Release 3 (10.1.3.1.0), OC4J instances belong to groups within the cluster topology, enabling you to perform group deployment, configuration, and administration operations across an Oracle Application Server cluster.
2. The Distributed Configuration Management (DCM) framework, used in prior releases of Oracle Application Server to replicate common configuration information across a cluster, is not included in the current release. This means that:
    a. Configuration using the dcmctl command line utility or Application Server Control Console is no longer supported.
    b. Cluster configurations must now be manually replicated in the opmn.xml file installed on each node within the cluster.
3. The ONS configuration file (ons.conf) is no longer used. ONS connection data is now set in the <notification-server> element within opmn.xml, the OPMN configuration file located in the ORACLE_HOME/opmn/conf directory on each node containing an OC4J or Oracle HTTP Server instance.

4. Each node is no longer required to be manually configured to connect to every other node in the cluster.

Oracle Application Server Clustering Process Overview

For every instance (ORACLE_HOME) that is part of the cluster, do the following steps.

1. Install PIA onto this instance.

   Note: Make sure same application name is specified during PIA Install on each instance.

2. Configure Oc4jRoutingID in mod_oc4j.conf file.

3. Setting up the Cluster

   a. Configuring Dynamic Node Discovery Using Multicast.

   b. Configuring Static Node-to-Node Communication.

Configure Oc4jRoutingID in mod_oc4j.conf file

1. Open $ORACLE_HOME\Apache\Apache\conf\mod_oc4j.conf file.

2. Based on the type of PIA install, modify the line **Oc4jRoutingID rid-home** to be either **rid-home,rid-<application-name>** or **rid-home,rid-PIA_<application-name>**. For example, if the application name is PeopleSoft

   a. For Single Component Server, update the line to **Oc4jRoutingID rid-home,rid-PeopleSoft.**

   b. For Multi Component Server, update the line to **Oc4jRoutingID rid-home,rid-PIA_PeopleSoft**.

   Note: Make sure there is no space between the entries

Configuring Dynamic Node Discovery Using Multicast

Dynamic node discovery is the most straightforward clustering configuration. In this model, each ONS node broadcasts a simple multicast message announcing its presence, enabling nodes within the cluster to dynamically discover one another.

Each ONS maintains its own map of the cluster topology. When a new ONS is added to the cluster, each existing ONS adds the new node and its connection information to its map. At the same time, the new ONS adds all of the existing nodes to its map. Alternatively, when an ONS is removed from the cluster, the maps for the remaining nodes are updated with this change.

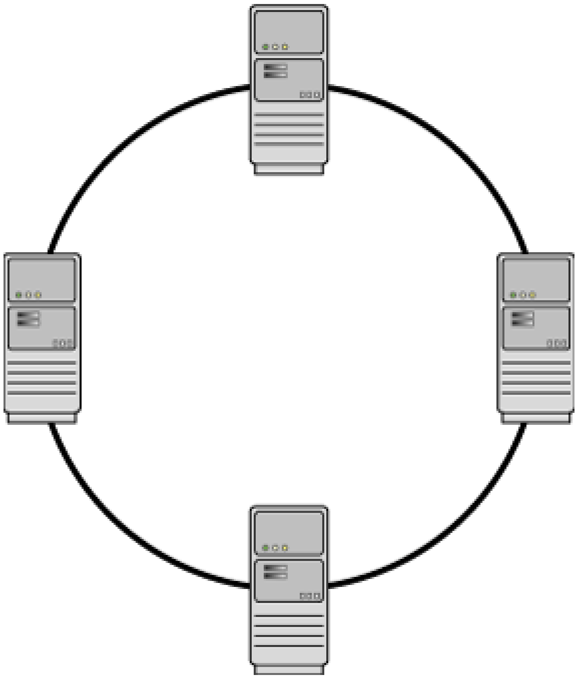


Figure: Dynamic Discovery Model

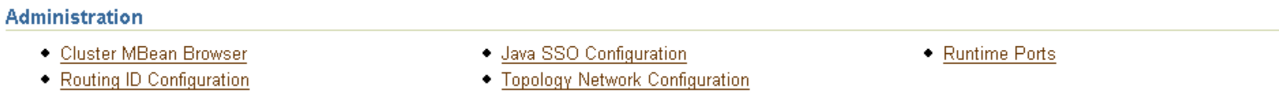
Because multicast messages may be restricted by different network configurations dynamic node discovery may be an option only for ONS nodes that are on the same subnet.

Notes:

- All nodes within the topology must be configured to use the same multicast address and port.
- The multicast address must be within the valid address range, which is 224.0.1.0 to 239.255.255.255. Ideally, multicast address and port assignments should be managed by your systems administration staff to avoid potential conflicts with other applications.

Adding instances to a cluster using multicast discovery

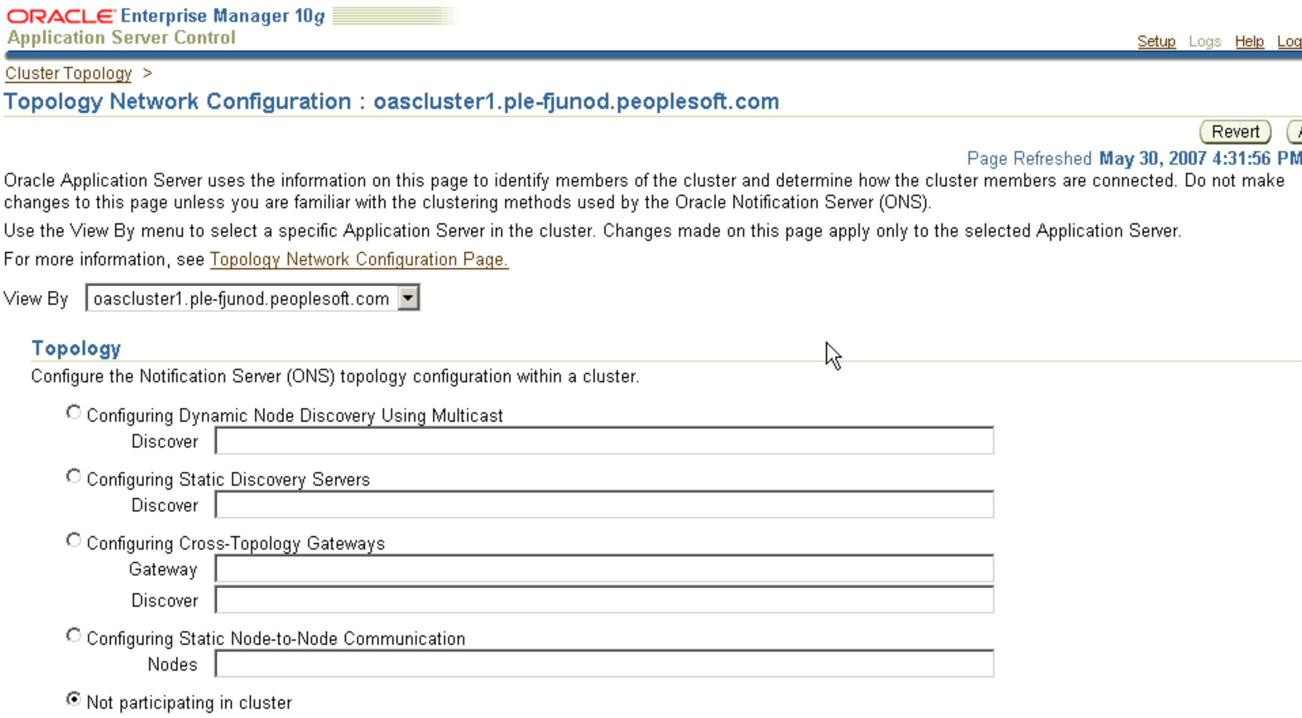
1. Using Application Server Control
    - Open your Web browser and enter the Application Server Control URL.
    - Log in to the Application Server Control, and scroll to the Administration section of the Cluster Topology page.



Administration

- Cluster MBean Browser
- Routing ID Configuration
- Java SSO Configuration
- Topology Network Configuration
- Runtime Ports

- Click **Topology Network Configuration** to display the Topology Network Configuration page.



Cluster Topology >

Topology Network Configuration : oascluster1.ple-fjunod.peoplesoft.com

Page Refreshed **May 30, 2007 4:31:56 PM**

Oracle Application Server uses the information on this page to identify members of the cluster and determine how the cluster members are connected. Do not make changes to this page unless you are familiar with the clustering methods used by the Oracle Notification Server (ONS).

Use the View By menu to select a specific Application Server in the cluster. Changes made on this page apply only to the selected Application Server.

For more information, see Topology Network Configuration Page.

View By oascluster1.ple-fjunod.peoplesoft.com

Topology

Configure the Notification Server (ONS) topology configuration within a cluster.

○ Configuring Dynamic Node Discovery Using Multicast
Discover

○ Configuring Static Discovery Servers
Discover

○ Configuring Cross-Topology Gateways
Gateway
Discover

○ Configuring Static Node-to-Node Communication
Nodes

● Not participating in cluster

- Note the following on the Topology Network Configuration page:

  i. The current application server instance is selected in the **View By** field.

  ii. The fields in the Topology section of the page are empty. This indicates that the application server instance does not belong to a cluster.

- Select **Configuring Dynamic Node Discovery Using Multicast** and enter a multicast address and port number in the **Discover** field.

  For example: 228.4.3.42:4242

  Note: In the configuration file, the multicast address must be preceded by an asterisk (*), but when you enter the address in this field, Application Server Control automatically includes the asterisk if you don't specify it here.

- Make a note of the multicast address and port, and then click **Apply**.

2. Using OPMN command line tool opmnctl

   The OPMN command-line tool opmnctl, supports a new *config topology* command that enables you to specify, update, or delete the multicast <discover> entry within opmn.xml.

   The opmnctl tool is installed in the ORACLE_HOME/opmn/bin directory of each node. The tool must be run individually on each node and will update only the opmn.xml file on that node.

   Open a command prompt and run the following commands:

   ```
   ORACLE_HOME/opmn/bin/opmnctl config topology update
   discover=”*multicastAddress:multicastPort”
   ```

   For example:

   ```
   ORACLE_HOME/opmn/bin/opmnctl config topology update
   discover=”*238.4.3.42:4242”
   ORACLE_HOME/opmn/bin/opmnctl reload
   ```

Configuring Static Node-to-Node Communication

The static configuration model is essentially the same mechanism used in Oracle Application Server 10.1.2 and 9.0.4. It continues to be supported primarily to provide backward compatibility with these earlier releases.

-

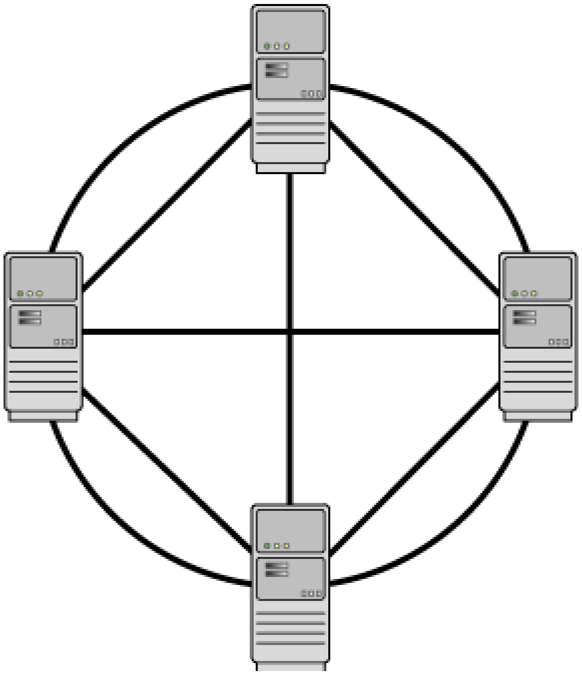


Figure: Static Node-to-Node model

In this configuration, a node list containing the host address and ONS remote listener port for each node in the cluster is supplied. Prior to Oracle Application Server Release 3 (10.1.3.1.0), when ONS configuration data was integrated into opmn.xml, this configuration would have been set in the ons.conf configuration file.

Define the host address and the ONS remote listener port - specified within the <port> sub element of <notification-server> - for each node in the cluster within the <nodes> sub element. Separate each node from the next with a comma.

For example:

```
<opmn>
 <notification-server>
  <port local="6101" remote="6202" request="6004"/>
  <ssl ... />
  <topology>
   <nodes list="node1-sun:6201,node2-sun:6202"/>
  </topology>
 </notification-server>
 ...
```

```
</opmn>
```

Supply the same list for each node in the cluster; each ONS instance will identify itself in the list and ignore that entry.

Note: The opmn.xml file must be reloaded by running "opmnctl reload" for changes to take effect in the OPMN runtime. Run the following command on the affected node to reload opmn.xml.

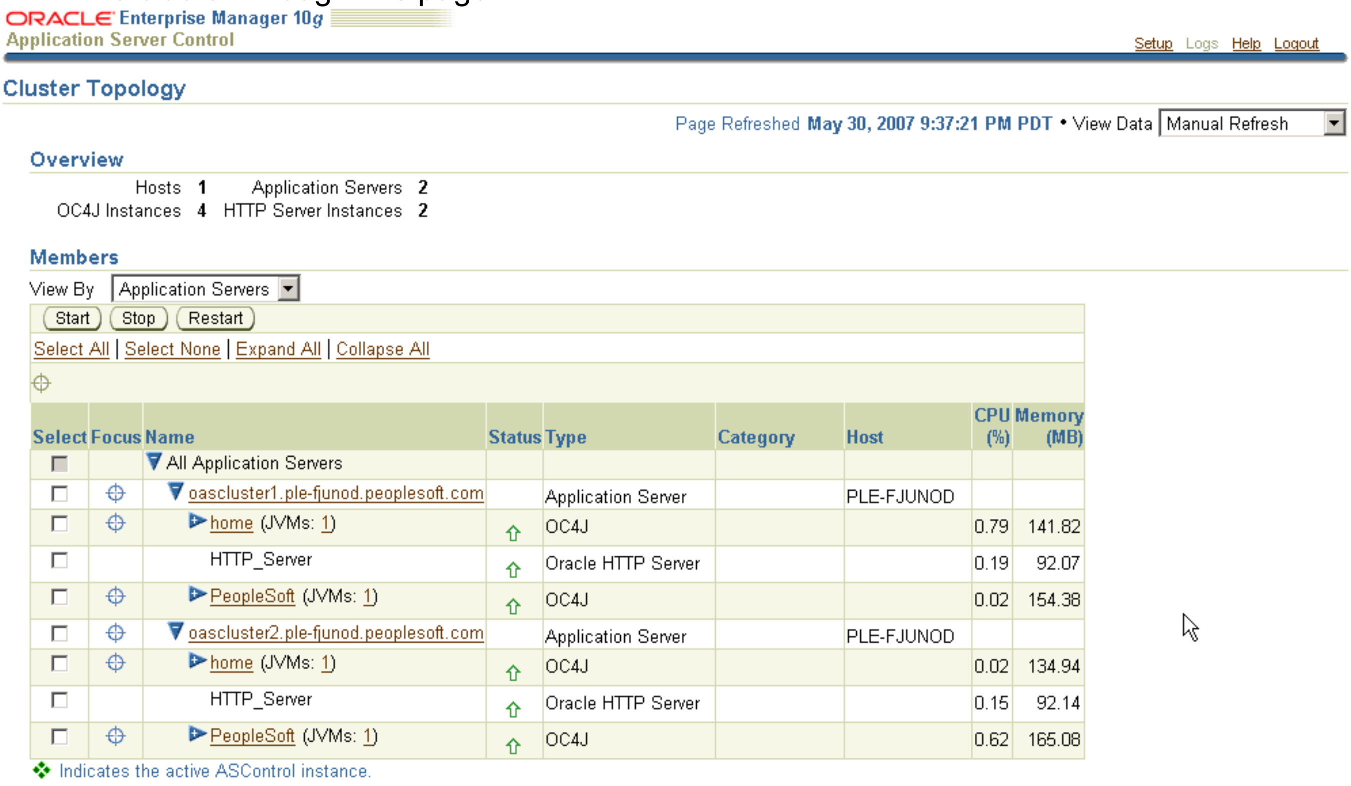
Viewing the Status of a Cluster

You can view the current status of the Oracle Application Server components within a cluster, using either opmnctl or Application Server Control Console.

- Viewing Cluster Status in Application Server Control Console

  Click the Cluster Topology link in the upper left corner of the Application Server Control Console home page.

  The resulting page displays each Oracle Application Server instance that is active within the cluster, as well as the active applications on each instance. You can access an instance or a deployed application within the cluster through this page.



ORACLE Enterprise Manager 10g
Application Server Control

Cluster Topology

Page Refreshed May 30, 2007 9:37:21 PM PDT • View Data Manual Refresh

Overview

Hosts	1	Application Servers	2
OC4J Instances	4	HTTP Server Instances	2

Members

View By Application Servers

Select	Focus	Name	Status	Type	Category	Host	CPU (%)	Memory (MB)
□		▼ All Application Servers						
□	⊕	▼ oascluster1.ple-fjunod.peoplesoft.com		Application Server		PLE-FJUNOD		
□	⊕	▶ home (JVMs: 1)	⇧	OC4J			0.79	141.82
□		HTTP_Server	⇧	Oracle HTTP Server			0.19	92.07
□	⊕	▶ PeopleSoft (JVMs: 1)	⇧	OC4J			0.02	154.38
□	⊕	▼ oascluster2.ple-fjunod.peoplesoft.com		Application Server		PLE-FJUNOD		
□	⊕	▶ home (JVMs: 1)	⇧	OC4J			0.02	134.94
□		HTTP_Server	⇧	Oracle HTTP Server			0.15	92.14
□	⊕	▶ PeopleSoft (JVMs: 1)	⇧	OC4J			0.62	165.08

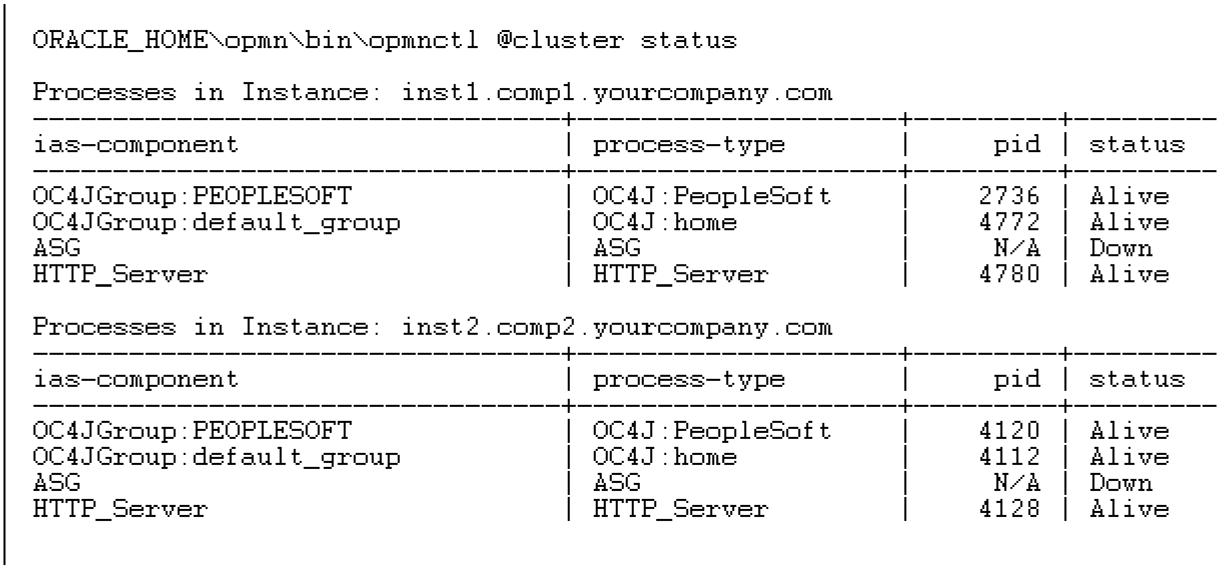
❖ Indicates the active ASControl instance.

- Viewing Cluster Status with opmnctl

  You can check the status of the cluster using opmnctl on any Oracle Application Server node within the cluster.

  ORACLE_HOME/opmn/bin/opmnctl @cluster status

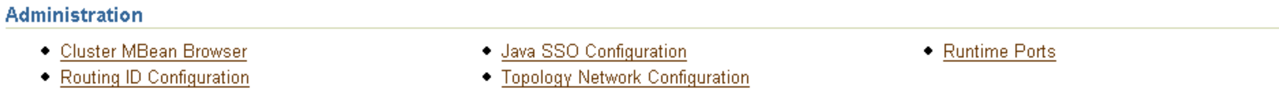
  The output shows the status of the components installed in each active Oracle Application Server instance within the cluster:



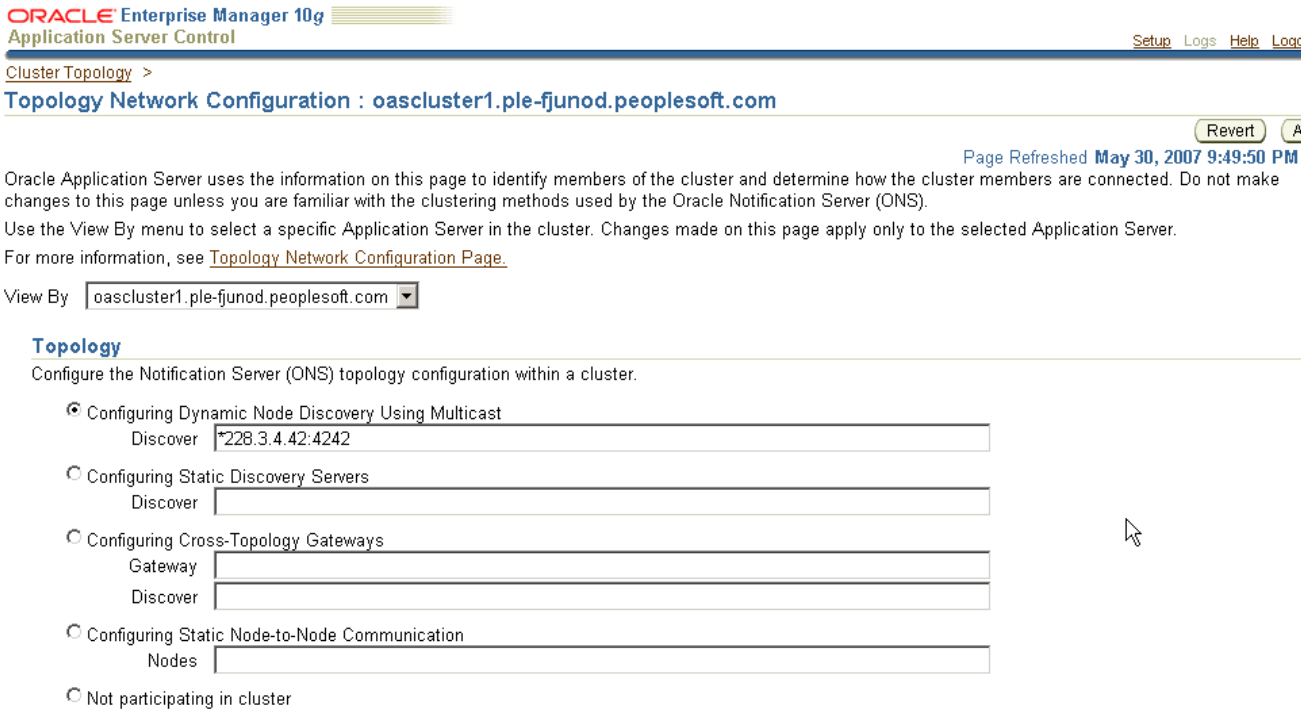
```
ORACLE_HOME\opmn\bin\opmnctl @cluster status

Processes in Instance: inst1.comp1.yourcompany.com
---------------------------------+--------------------+---------+---------
ias-component                    | process-type       |     pid | status
---------------------------------+--------------------+---------+---------
OC4JGroup:PEOPLESOFT             | OC4J:PeopleSoft    |    2736 | Alive
OC4JGroup:default_group          | OC4J:home          |    4772 | Alive
ASG                              | ASG                |     N/A | Down
HTTP_Server                      | HTTP_Server        |    4780 | Alive

Processes in Instance: inst2.comp2.yourcompany.com
---------------------------------+--------------------+---------+---------
ias-component                    | process-type       |     pid | status
---------------------------------+--------------------+---------+---------
OC4JGroup:PEOPLESOFT             | OC4J:PeopleSoft    |    4120 | Alive
OC4JGroup:default_group          | OC4J:home          |    4112 | Alive
ASG                              | ASG                |     N/A | Down
HTTP_Server                      | HTTP_Server        |    4128 | Alive
```

Removing an instance from a cluster

- Using Application Server Control
  - Open your Web browser and enter the Application Server Control URL.
  - Log in to the Application Server Control, and scroll to the Administration section of the Cluster Topology page.



  - Click **Topology Network Configuration** to display the Topology Network Configuration page.
  - In the Topology Network Configuration page, select an application server instance in the **View By** field that needs to be removed from the cluster



Topology Network Configuration : oascluster1.ple-fjunod.peoplesoft.com

Page Refreshed May 30, 2007 9:49:50 PM

Oracle Application Server uses the information on this page to identify members of the cluster and determine how the cluster members are connected. Do not make changes to this page unless you are familiar with the clustering methods used by the Oracle Notification Server (ONS).

Use the View By menu to select a specific Application Server in the cluster. Changes made on this page apply only to the selected Application Server.

For more information, see Topology Network Configuration Page.

View By oascluster1.ple-fjunod.peoplesoft.com

Topology

Configure the Notification Server (ONS) topology configuration within a cluster.

● Configuring Dynamic Node Discovery Using Multicast
Discover *228.3.4.42:4242

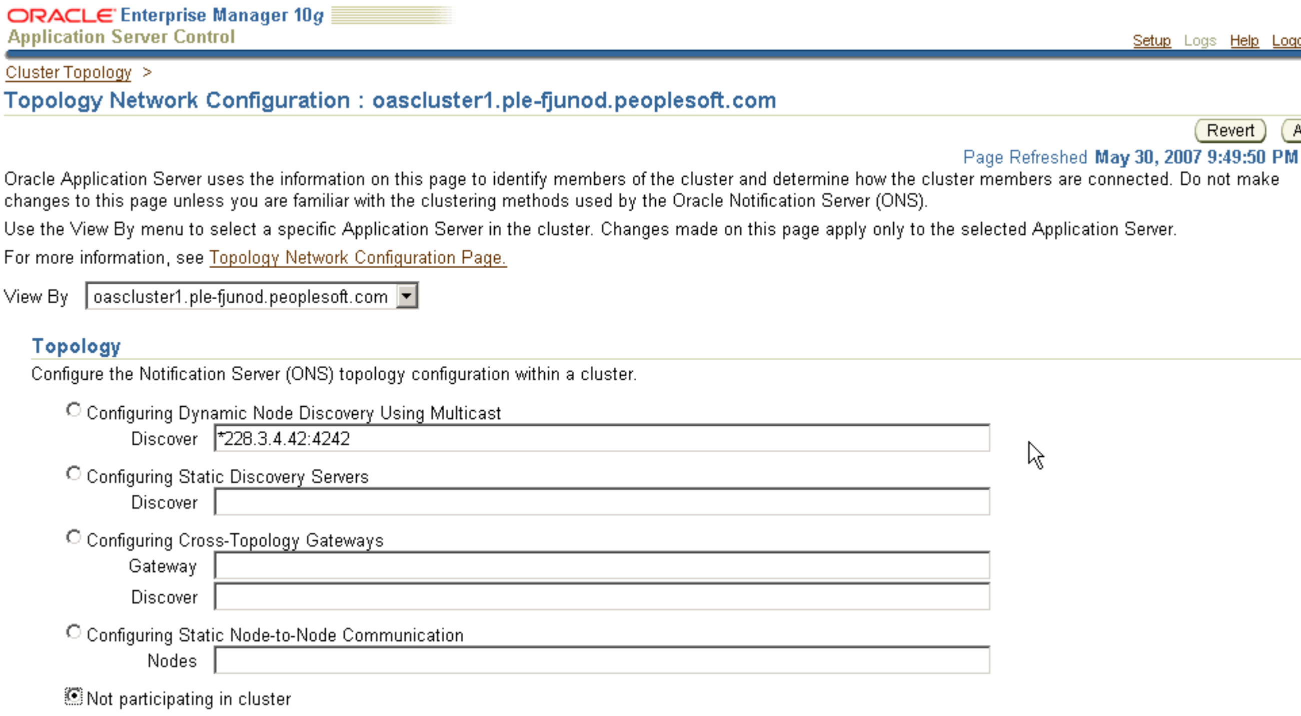
○ Configuring Static Discovery Servers
Discover

○ Configuring Cross-Topology Gateways
Gateway
Discover

○ Configuring Static Node-to-Node Communication
Nodes

○ Not participating in cluster

- Click **Not participating in cluster** option



Topology Network Configuration : oascluster1.ple-fjunod.peoplesoft.com

Page Refreshed May 30, 2007 9:49:50 PM

Oracle Application Server uses the information on this page to identify members of the cluster and determine how the cluster members are connected. Do not make changes to this page unless you are familiar with the clustering methods used by the Oracle Notification Server (ONS).

Use the View By menu to select a specific Application Server in the cluster. Changes made on this page apply only to the selected Application Server.

For more information, see Topology Network Configuration Page.

View By oascluster1.ple-fjunod.peoplesoft.com

Topology

Configure the Notification Server (ONS) topology configuration within a cluster.

○ Configuring Dynamic Node Discovery Using Multicast
Discover *228.3.4.42:4242

○ Configuring Static Discovery Servers
Discover

○ Configuring Cross-Topology Gateways
Gateway
Discover

○ Configuring Static Node-to-Node Communication
Nodes

● Not participating in cluster

- Click **Apply**

- Using OPMN command line tool opmnctl

The delete command removes the <discover> element from opmn.xml, effectively removing the node from the cluster. If the <topology> element contains no other sub elements, it will be removed as well.

ORACLE_HOME/opmn/bin/opmnctl config topology delete discover
ORACLE_HOME/opmn/bin/opmnctl reload

CONFIGURING A CISCO CSS LOADBALANCER

This section applies to Advanced and Generic Webserver Clustering architecture only.

Getting Started

- Cisco CSS switch uses non-standard serial pin-out configuration for console connection. It is also different from Cisco's standard rollover cable. Only use the adapters included with the switch. Use the following terminal setting – 8 bit, No Parity, 1 stop bit, 9600 baud.
- For a fresh unit use – Username: *admin* Password: *system*
- At the prompt "No startup-config was found, continue with the setup script [y/n]?" select 'n' to exit setup script.
- Setup prompt for the unit:

  ```
  CS150# prompt lb-1
  lb-1#
  ```
- Save user specific environment:

  ```
  lb-1# save_profile
  ```
- Go to config mode:

  ```
  lb-1# config
  lb-1(config)#
  ```
- To configure port 1's IP address choose circuit 1:

  ```
  lb-1(config)# circuit VLAN 1
  lb-1(config-circuit[VLAN 1])#
  ```
- Configure port 1's IP address:

  ```
  lb-1(config-circuit[VLAN 1])# ip address 10.0.0.5
  255.255.255.0
  lb-1(config-circuit[VLAN 1-10.0.0.5])#
  ```
- Put some description for this VLAN:

  ```
  lb-1(config-circuit[VLAN 1])# description "Outside
  Network"
  ```
- Configure the default route :

  ```
  lb-1(config)# ip route 0.0.0.0 0.0.0.0 10.0.0.1
  ```

- Disable spanning tree support since this is a loadbalancer and not a Layer 3 router :

  ```
  lb-1(config)# bridge spanning-tree disabled
  ```

- Configure the redundancy unit similar to the previous steps with the following changes: change `lb-1` to `lb-2`, change IP `10.0.0.5` to `10.0.0.6`

At this point basic networking is complete and you should be able to login to the unit from the network to complete the installation.

Optional Setup for NAT

To setup NAT you have to setup additional VLAN circuits.

- To configure port 2:

  ```
  lb-1(config)# interface ethernet-2
  lb-1(config-if[ethernet-2])# bridge vlan 2
  ```

  (NOTE: vlan is lowercase in this syntax)

- To configure port 2’s IP address now:

  ```
  lb-1(config-if[ethernet-2])# circuit VLAN 2
  lb-1(config-circuit[VLAN 2])# description “Inside
  Network”
  lb-1(config-circuit[VLAN 2])# ip address 10.0.0.1
  255.255.255.0
  Create ip interface <10.0.0.1>, [y/n]? y
  lb-1(config-circuit-ip[VLAN 2-10.0.0.1])#
  ```

Create Service for Webinstance

Each webinstance should be configured as a “service” in CSS terms.

- To configure webinstance1 as a service:

  ```
  lb-1(config)# service web-inst1
  Create service <web-inst1>, [y/n]? y
  lb-1(config-service[web-inst1])# ip address
  10.0.0.10
  lb-1(config-service[web-inst1])# port 5010
  lb-1(config-service[web-inst1])# active
  ```

- Setup keep-alive (this will ping the service at an interval to determine availability) :

  ```
  lb-1(config-service[web-inst1])# keepalive type http
  lb-1(config-service[web-inst1])# keepalive method
  get
  lb-1(config-service[web-inst1])# keepalive uri
  “/index.html”
  lb-1(config-service[web-inst1])# keepalive frequency
  15
  ```

- Check configuration with:

  ```
  lb-1(config)# show service web-inst1
  ```
- Create a similar services for SSL e.g. `web-inst1-ssl`
- Create similar services for other webinstances e.g. `web-inst2(-ssl),web-inst3(-ssl),web-inst4(-ssl)`

Create VIP

Each VIP should be configured as a "content rule" in CSS terms.

- To configure a content rule you must first setup an "owner":

  ```
  lb-1(config)# owner PS-WEB
  Create owner <PS-WEB>, [y/n]? y
  lb-1(config-owner[PS-WEB])#
  ```
- Once the owner is created setup a content rule:

  ```
  lb-1(config-owner[PS-WEB])# content vip-1
  Create content <vip-1>, [y/n]? y
  lb-1(config-owner-content[PS-WEB-vip-1])#
  ```
- Associate the IP address with the created VIP (content rule):

  ```
  lb-1(config-owner-content[PS-WEB-vip-1])# vip
  address 10.0.0.100
  ```
- Assign protocol etc. with the created VIP (content rule):

  ```
  lb-1(config-owner-content[PS-WEB-vip-1])# protocol
  tcp
  lb-1(config-owner-content[PS-WEB-vip-1])# url "/*"
  lb-1(config-owner-content[PS-WEB-vip-1])# balance
  aca
  ```
- Setup sticky (persistence):

  For HTTP

  ```
  lb-1(config-owner-content[PS-WEB-vip-1])# port 80
  lb-1(config-owner-content[PS-WEB-vip-1])# advanced-
  balance arrowpoint-cookie
  ```

  For HTTPS

  ```
  lb-1(config-owner-content[PS-WEB-vip-1])# port 443
  lb-1(config-owner-content[PS-WEB-vip-1])#
  application ssl
  lb-1(config-owner-content[PS-WEB-vip-1])# advanced-
  balance ssl
  ```

  NOTE- for Pre 8.17 Tools release running Portal and using Generic Webserver clustering, advanced-balance must be set to **"sticky-srcip-dstport"** for both HTTP and HTTPS.
- Assign webinstances (services) with the created VIP (content rule):

```
lb-1(config-owner-content[PS-WEB-vip-1])# add
service web-inst1
lb-1(config-owner-content[PS-WEB-vip-1])# add
service web-inst2
lb-1(config-owner-content[PS-WEB-vip-1])# add
service web-inst3
lb-1(config-owner-content[PS-WEB-vip-1])# add
service web-inst4
```

- Set content rule active:

```
lb-1(config-owner-content[PS-WEB-vip-1])# active
```

Setup Redundancy

Cisco CSS handles redundancy using a modified VRRP. To setup redundancy between the two units a private network must be setup between the two. Up to now the two units have been set with independent IP addresses. VIP and servers have been configured only on the master (lb-1) unit.

- To prevent problematic bridging loops do not connect the units to each other yet.

- To setup the private network, configure any port, say 12:

```
lb-1(config)# interface ethernet-12
lb-1(config-if[ethernet-12])# bridge vlan 3
```

(NOTE: vlan is lowercase in this syntax)

- Then configure port 12 with an RFC1918 IP address now. This network is only used by the units to do a health check between the units:

```
lb-1(config-if[ethernet-12])# circuit VLAN 3
lb-1(config-circuit[VLAN 3])# description
"Redundancy Network"
lb-1(config-circuit[VLAN 3])# ip address 172.16.0.1
255.255.255.0
Create ip interface <172.16.0.1>, [y/n]? y
lb-1(config-circuit-ip[VLAN 3-172.16.0.1])#
```

- Connect the two units with a crossover cable at this point

- Setup master redundancy on unit lb-1:

```
lb-1(config)# ip redundancy master
```

- Synch up the setup between unit1 and unit2:

```
lb-1(config)# app
lb-1(config)# app session 172.16.0.2
```

- To setup the standby unit2 it must have the inactive IP address of 10.0.0.6. To do so make the changes via serial console:

```
lb-2(config)# app
lb-2(config)# app session 172.16.0.1
```

- Setup redundancy on unit lb-2 (not the master):

  ```
  lb-2(config)# ip redundancy
  ```

- Configure redundancy to run between the switches:

  ```
  lb-2(config)# circuit VLAN 3
  lb-2(config-circuit[VLAN 3])# ip address 172.16.0.2
  255.255.255.0
  Create ip interface <172.16.0.2>, [y/n]? y
  lb-2(config-circuit-ip[VLAN 3-172.16.0.2])#
  ```

- The next step is to configure individual interfaces for redundancy. To do so setup redundancy for VLAN 1 (and VLAN 2 if you are using NAT):

  ```
  lb-2(config)# circuit VLAN 1
  lb-2(config-circuit[VLAN 1])# redundancy
  ```

- At this point change the IP address of the standby unit to be the same as the IP address of the active unit:

  ```
  lb-2(config-circuit[VLAN 1])# no ip address 10.0.0.6
  Delete ip interface <10.0.0.5>, [y/n]? y
  lb-2(config-circuit[VLAN 1])# ip address 10.0.0.5
  255.255.255.0
  Create ip interface <10.0.0.5>, [y/n]? y
  lb-2(config-circuit-ip[VLAN 1-10.0.0.5])#
  ```

- From this point onwards whenever a new change is performed on the active unit it should be synched to the standby unit with the following command:

  ```
  lb-1# script play commit_redundancy “172.16.0.2 –a”
  ```

Check overall configuration

- Save configuration with:

  ```
  lb-1# save_config
  ```

- To check all configuration settings run:

```
lb-1# show run
```

CREATING LOGICAL IP ADDRESSES (IP ALIASES)

This section applies to Simple and Advanced WebLogic Clustering architecture only.

You will need to create IP Aliases if you are using:

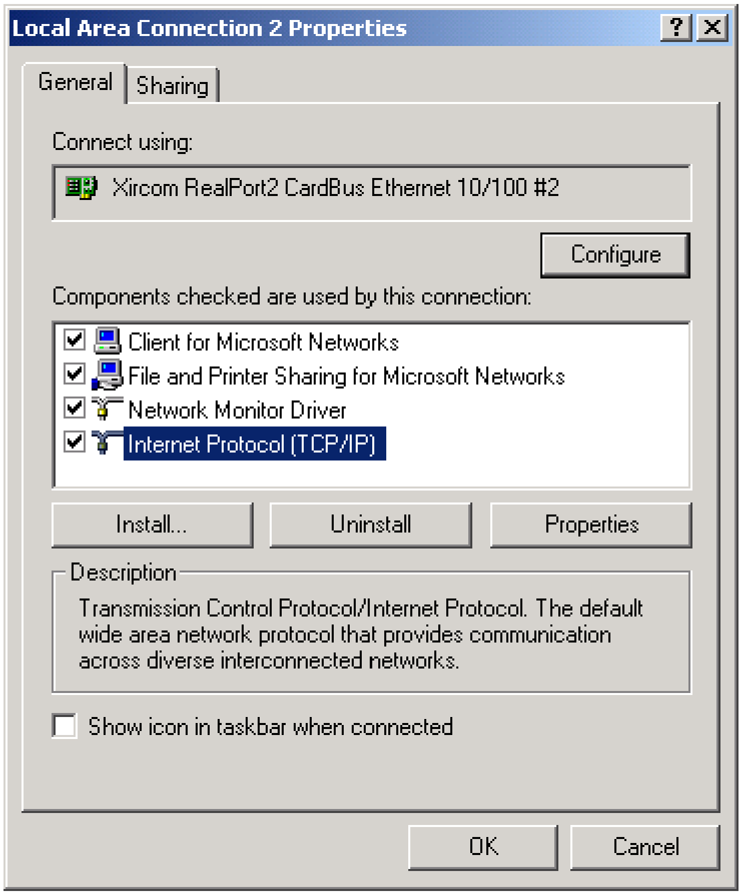
WebLogic proxy **and**

You need to install multiple WebLogic instances on a host **and**

You do not have dedicated network interfaces for each of the WebLogic instances being installed

IP Aliases on Windows 2000

1. Go into the Control Panel Window and select Network and Dialup Connections.
2. Right Click on the Local Area Connection icon which is associated to the network interface where the IP aliases need to be created
3. Select Properties.
4. From the Local Area Connection Property dialog select Internet Protocol (TCP/IP) then select Properties



Connect using:

Xircom RealPort2 CardBus Ethernet 10/100 #2

Components checked are used by this connection:

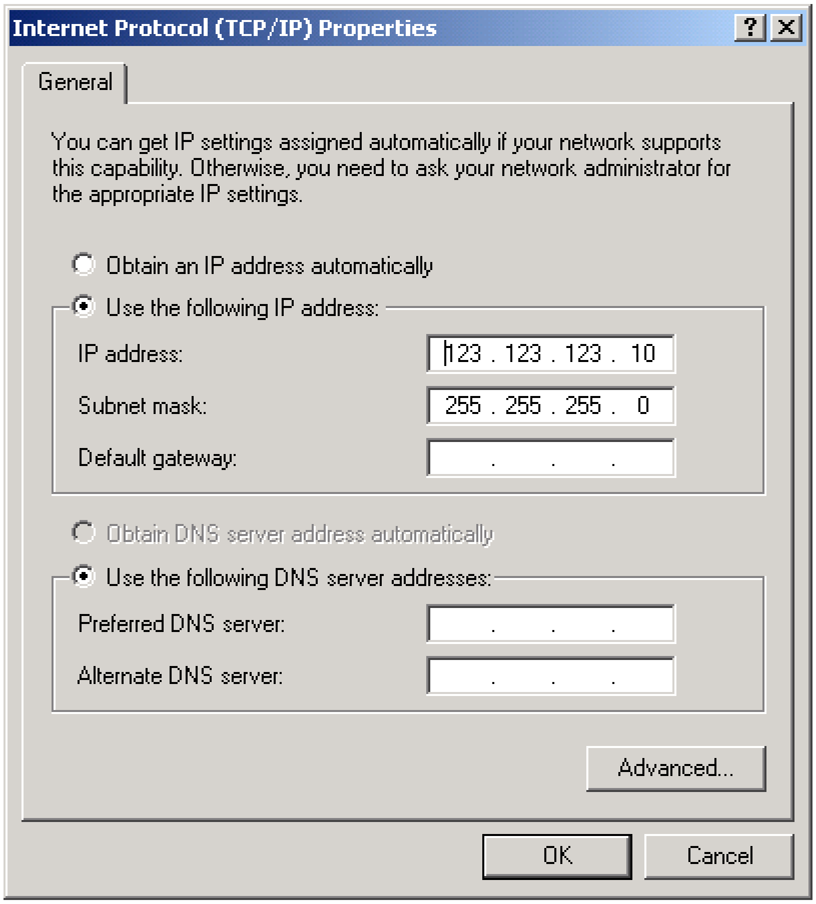
☑ Client for Microsoft Networks
☑ File and Printer Sharing for Microsoft Networks
☑ Network Monitor Driver
☑ Internet Protocol (TCP/IP)

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

☐ Show icon in taskbar when connected

5. Select Use The Following IP Address and add the primary IP address for this interface.



Internet Protocol (TCP/IP) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 123 . 123 . 123 . 10

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

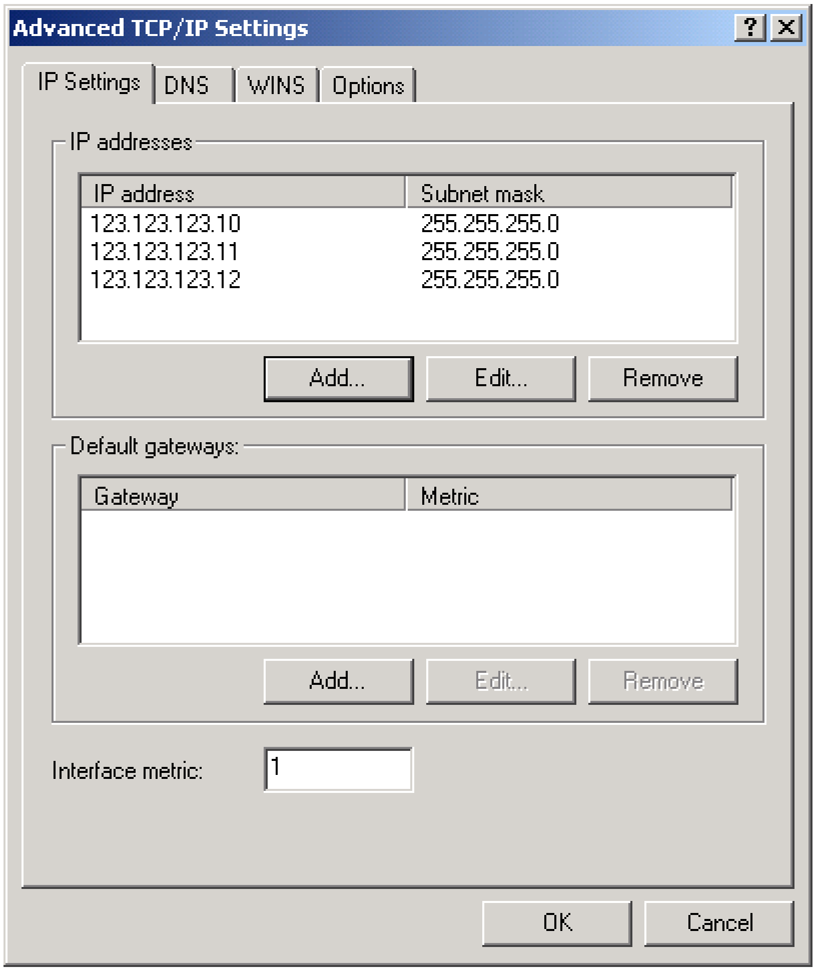
● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

6. Select Advanced…

7. From the Advanced TCP/IP Settings add the IP Aliases for this interface



IP address	Subnet mask
123.123.123.10	255.255.255.0
123.123.123.11	255.255.255.0
123.123.123.12	255.255.255.0

Gateway	Metric

Interface metric: 1

8. Once all the IP Aliases have been added select OK all the way out until all values are saved.

IP Aliases on Solaris

1. On Solaris use ifconfig to create IP Aliases. Do not attempt this on a production machine without testing it out on a isolated host.
2. Run ifconfig –a to see all the interfaces on the host.

```
st-sun03:$ ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu
8232
        inet 127.0.0.1 netmask ff000000
hme0:
flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST>
mtu 1500
        inet 123.123.123.10 netmask ffffff00
broadcast 123.123.123.255
st-sun03:$
```

3. As Super User run the command :

Solaris 2.7

```
st-sun03:# ifconfig hme0:1 123.123.123.11 netmask
255.255.255.0
st-sun03:# ifconfig hme0:2 123.123.123.12 netmask
255.255.255.0
```

Solaris 2.8

```
st-sun03:# ifconfig hme0 addif 123.123.123.11\24
st-sun03:# ifconfig hme0 addif 123.123.123.12\24
```

4. Run Run ifconfig –a to verify the IP Aliases created.

```
st-sun03:$ ifconfig -a
lo0: flags=849<UP,LOOPBACK,RUNNING,MULTICAST> mtu
8232
        inet 127.0.0.1 netmask ff000000
hme0:
flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST>
mtu 1500
        inet 123.123.123.10 netmask ffffff00
broadcast 123.123.123.255
hme0:1
flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST>
mtu 1500
        inet 123.123.123.11 netmask ffffff00
broadcast 123.123.123.255
hme0:2
flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST>
mtu 1500
        inet 123.123.123.12 netmask ffffff00
broadcast 123.123.123.255
st-sun03:$
```

Note: you will need to update /etc/rc.d files to make sure changes survive reboots

Note: if you need to delete an IP alias use the following command:

Solaris 2.7

```
st-sun03:# ifconfig  hme0:1 0 down
```

Solaris 2.8

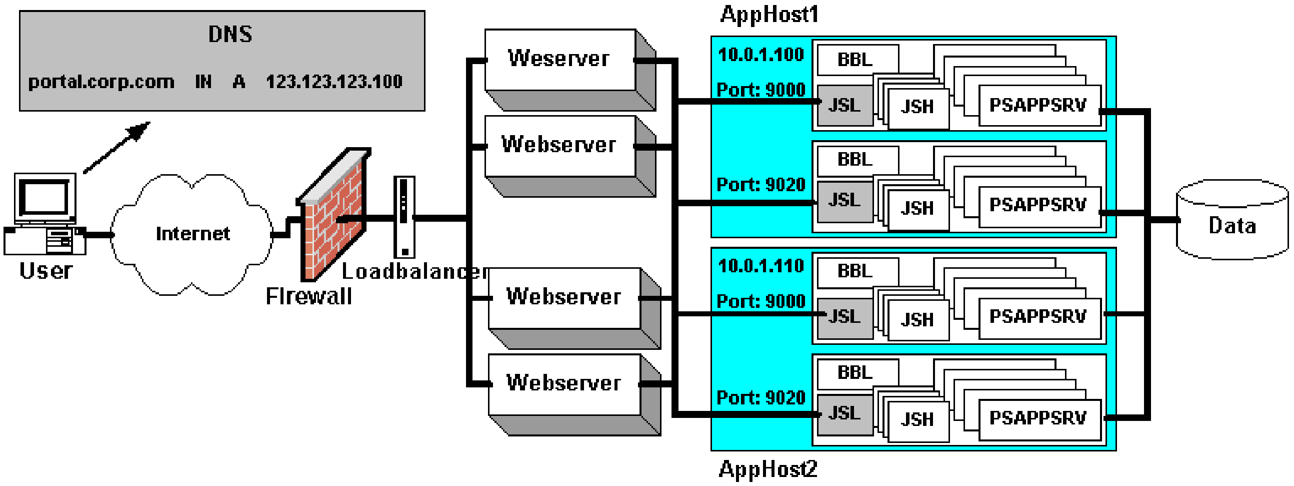
```
st-sun03:# ifconfig  hme0:1 down unplumb
```

Chapter 4 Application Server Clustering

High availability for application servers is provided out of the box for all PeopleSoft systems by using BEA Tuxedo's domain failover mechanism. Instead of clustering application servers high availability is provided by running multiple domains and using software loadbalancing and failover. Within a PeopleSoft system all application server processes belong to a domain. There can be multiple domains running on a host but domains do not span multiple hosts. Therefore each host will contain one or more application server domains. Within a domain there are multiple application server processes that handle service requests. The total number of application server processes in a domain is configurable and should be set according to the server's capacity. Service requests are load balanced across these server processes. If any one of these application server processes crash, Tuxedo will route any active call currently handled by the affected process to another available server process in the domain. It will also restart the failed server process if needed. This is application server recovery mechanism. The failure is handled transparently and no error is reported to the webserver (JOLT client).

All calls from a webserver are further distributed among all available domains. A domain is considered available when a network connection can be established to that domain. If a domain fails while a call is outstanding the call is restarted with another domain that is part of the failover configuration.

Domain level loadbalance and failure setup is documented in configuration.properties section of "*Web Administration*" PeopleBook. As a general high availability good practice create two domains per host if the number of appserver hosts is small. This will increase the complexity of administrating the system but will provide greater flexibility in terms of maintaining high availability. With two application server domains per host, unstable domains can be restarted with half the impact on system capacity. The architecture is shown below:



LOADBALANCING AND FAILOVER SETUP

In the instructions below `WL_HOME` is the directory where WebLogic is installed, "myserver" is the cluster name. The appssever load balance and failover configuration steps to follow are:

1. Create two domains per appserver host, e.g. On AppHost1 create a domain with the JOLT listener (JSL) running at IP port 9000 and the second domain with JOLT listener running at IP port 9020.
2. Edit `WL_HOME\myserver\psftdocs\peoplesoft8\configuration.properties`
   - Update the `psserver` field with a , (comma) separated list of all domain (JOLT) listeners that need to be load balanced or are part of the failover setup e.g.

     ```
     psserver=AppHost1:9000,AppHost1:9020,AppHost2:9000
     ,AppHost2:9020
     ```
3. Repeat step 2 for all webserver hosts.

ADDITIONAL SETUPS ON APPSERVERS

Besides connecting to the database the Application server may also connect to other systems depending on how the PeopleSoft system is configured. The following appserver components need additional high availability consideration.

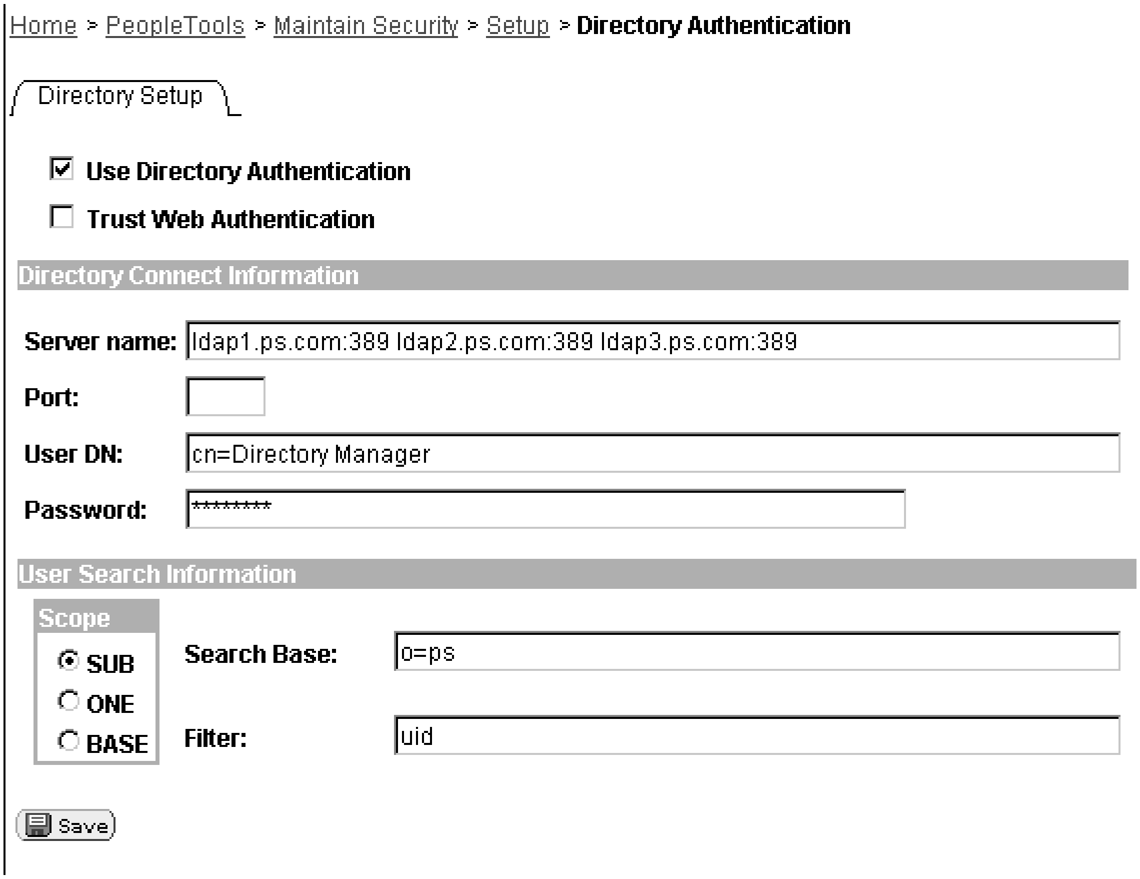
Security Manager

The security manager in the application server communicates with Directory Servers using Lightweight Directory Access Protocol (LDAP) to authenticate end users and manage their system access privileges. Since the communication to the Directory Server is LDAP customers can use any LDAP version 3 compliant server (such as Netscape Directory Server or Microsoft Active Directory) instead of NDS.

In general Directory Servers use replication to achieve scalability and high availability. PeopleSoft Directory access architecture has out-of-the-box support for multi-server directories. This allows PeopleTools security manager to work with enterprise class, globally distributed directory implementations. Features such as automatic fail-over in the event that a directory server is unavailable and referral handling across distributed directories (e.g. Multi-Domain Active Directory implementations and/or partitioned NDS trees) are supported.

Failover

Failover is configured with a space separated list of LDAP servers in the security manager connect sting. The following instructions show how to configure PeopleSoft security manager for failover:



Home > PeopleTools > Maintain Security > Setup > **Directory Authentication**

Directory Setup

☑ **Use Directory Authentication**

□ **Trust Web Authentication**

Directory Connect Information

Server name:	ldap1.ps.com:389 ldap2.ps.com:389 ldap3.ps.com:389
Port:	
User DN:	cn=Directory Manager
Password:	*********

User Search Information

Scope: ● SUB ○ ONE ○ BASE

Search Base:	o=ps
Filter:	uid

Loadbalancing

Random Server

To load balance authentication requests across multiple directory servers PeopleCode customizations are needed. The following PeopleCode implements a simple server selection algorithm for loadbalancing across multiple servers. The following function just randomly picks an element from a space delimited string list and moves it to the head. To use it just need add one line to the getLDAPConfig() function in signon ppc as follows :

```
*////////////////////////////////////////////////////
////////////////////////////////////////////////////
///////
  getLDAPConfig() reads the LDAP Business Interlink
settings from PS_DIRECTORYSETUP
////////////////////////////////////////////////////
////////////////////////////////////////////////////
//////*/
```

```
Function getLDAPConfig()

   &sql_ldapMap = CreateSQL("Select LDAPSERVERNAME,
LDAPUSERDN, LDAPUSERPSWD, LDAPPORT, LDAPSCOPE,
LDAPSRCHBASE, LDAPFILTER from PS_DIRECTORYSETUP");
   &ret = &sql_ldapMap.Fetch(&host, &cnctDN,
&cnctPWD, &port, &tmpscope, &base, &authAttr);

   Evaluate &tmpscope
   When 1
      &scope = "base"; /* search base entry*/
   When 2
      &scope = "one"; /* search one level */
   When 3
      &scope = "sub"; /* search the subtree */
   End-Evaluate;

&host = randomHost(&host);     /* <<<<<<<<<<<
new line   <<<<<<<<<<<<<<< */

   &bConfigRead = True;
End-Function;




Function randomHost(&str As string) Returns string;

   &s = &str;
   &not_done = True;
   &count = 1;
   &posArray = CreateArrayRept(0, 0);
   &posArray.Len = 0;

   While &not_done
      &pos = Find(" ", &s);
      If &pos = 0 Then
         &not_done = False;
      Else
         &s = Right(&s, Len(&s) - &pos);
         If &count = 1 Then
            &posArray.push(&pos);
         Else
            &posArray.push(&pos + &posArray [&count
- 1]);
         End-If;
         &count = &count + 1;
      End-If;
   End-While;

   &random = Int(Rand() * &count);

   If &random = 1 Then
```

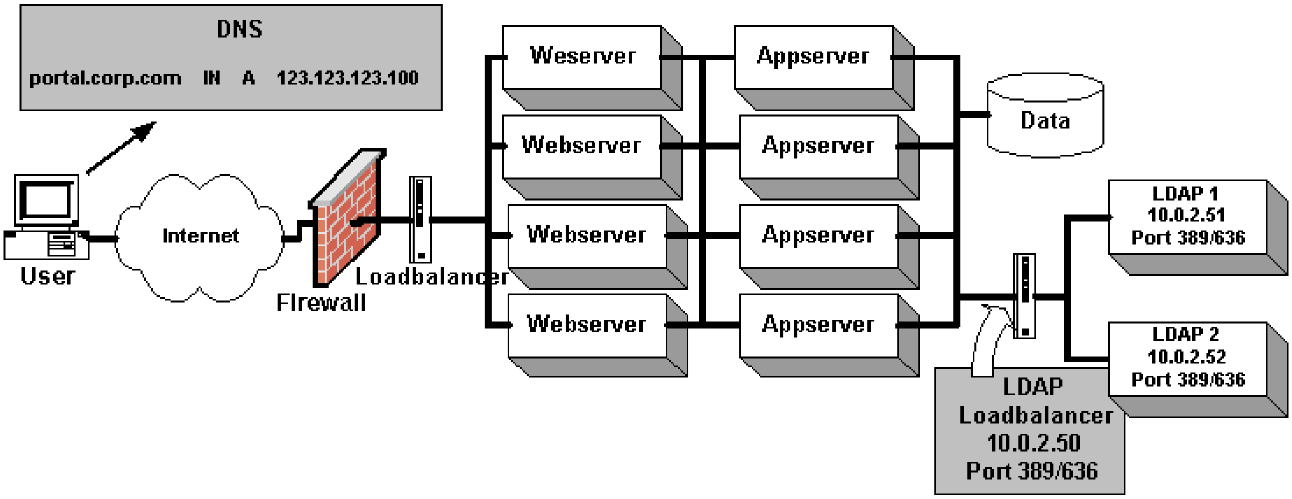
```
      Return &str;
   End-If;

   If &random = &count Then
      &result = Right(&str, Len(&str) - &posArray
[&posArray.Len]);
      &result = &result | " " | Left(&str, &posArray
[&posArray.Len]);
   Else
      &result = Substring(&str, &posArray [&random -
1], &posArray [&random] - &posArray [&random - 1]);
      &result = &result | " " | Left(&str, &posArray
[&random - 1]); /* preserves the trailing space */
      &result = &result | Right(&str, Len(&str) -
&posArray [&random]);
   End-If;

   Return &result;

End-Function;
```

Advanced Loadbalancing and Failover

For highly scalable and available systems PeopleSoft recommends a HW loadbalancer. No PeopleCode customization is needed in this case and the configuration provides both loadbalancing and failover. To configure a HW loadbalancer:

- Create virtual IP for the directory servers
- Setup system for TCP loadbalance
- Enable sticky based on client IP address
- Configure PeopleSoft as if only one LDAP server is at Virtual IP address
- Setup failover for the loadbalancer

The architecture for a loadbalanced security manager is shown below:



Directory Server Setup

A PeopleSoft system can use one of the following Directory Service systems:

Novell NDS

Novell NDS uses replication and clustering for high availability. Novell recommends three replicas on three separate servers. It is also possible to have multiple replicas (copies) of the Directory on one server. There can be three different kind of replicas: Master Replica (by default this is the first server that is built), Read/Write replica, Read-Only Replica and Subordinate Reference Replica (used by the system only). There is only one master replica per partition at any given time. However, an administrator can promote/demote servers between these replication roles as required. A master replica must be available on the network for NDS to perform operations such as creating a new replica or creating a new partition. For highest availability it is recommended that the replica master be located on a clustered NetWare server. Other replicas can reside on any other NDS supported OS platform. PeopleSoft security manager can run with either of the first three replicas for authenticating users. NDS uses partitioning for scalability. For a small system (i.e. few nodes in the directory connected over a freely routed LAN) the default single partition will suffice. For systems with complicated network topology (i.e. one or more low bandwidth links) the replication scheme will require careful planning, partitioning and forming a replica ring using the Replica Advisor tool. Instructions for configuring NDS systems are available at the following site:

http://www.novell.com/documentation/

Netscape Directory Server

Netscape Directory server use Single Supplier and multiple Consumer replication scheme. In this scheme, every directory object must be mastered by one and only one Directory Server. This mastering Directory Server is called the supplier server because it supplies the object to other servers. Servers that receive directory objects from supplier servers are called consumer server. Replication include provision for whole tree replication, subtree replication, cascading replication, and multiple subtree replication. For a small system (i.e. few nodes in the directory connected over a freely routed LAN) the default whole tree replication will suffice. For systems with complicated network topology (i.e. one or more low bandwidth links) the replication scheme will require careful planning, creating subtrees and selecting proper replication schemes. Instructions for configuring Netscape Directory server is available at the following site:

http://developer.netscape.com/docs/manuals/index.html?content=directory/41/de/contents.htm

Windows 2000 Active Directory

Windows 2000 uses Multi-master replication protocol to achieve scalability and high availability. In this scheme a highly available system will contain more than one Global Catalog (GC) and Domain Controller (DC) in the Windows 2000 forest. In multi-master mode updates are allowed on all available GCs or DCs and the system uses Knowledge Consistency Checker (KCC) to keep them in synch. For a small system (i.e. few nodes in the forest connected over a freely routed LAN) the default replication mechanism will suffice. For systems with complicated network topology (i.e. one or more low bandwidth links) the replication scheme will require careful planning of multiple sites and routes connected with weighted bridges. The following link provides instructions on how to optimize replication for a large LAN.

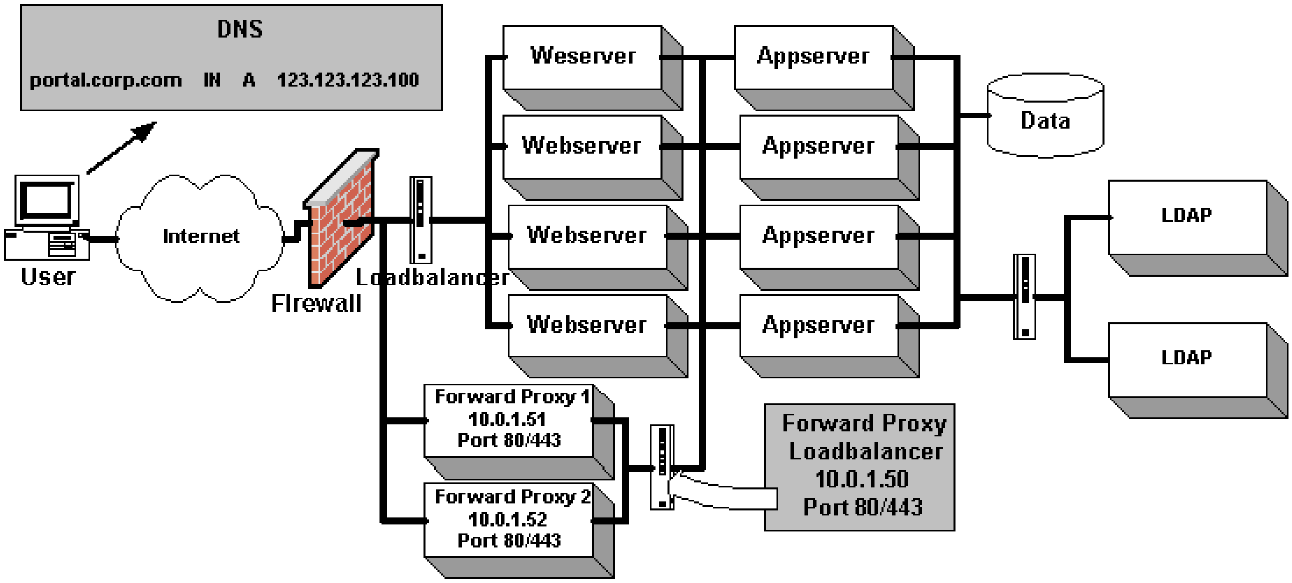
http://support.microsoft.com/default.aspx?scid=kb;EN-US;q244368

Application Messaging & Business Interlinks

PeopleSoft Application Messaging is a server-based architecture that allows PeopleSoft applications to publish messages in response to the invocation of business events within the application server. These messages are published in XML format and delivered to subscribing systems over a secure HTTP connection. It is not uncommon for a PeopleSoft system to publish across firewall boundaries via forward Proxy servers. If that is the case we need to make sure that a highly available forward proxy capability be available for the application servers to use for publishing the information.

PeopleSoft Business Interlinks architecture is a plug-in framework for PeopleSoft applications to invoke third-party APIs over the internet. Different vendors support different methods for invoking their APIs—including object technologies such as COM, CORBA, EJB; programming language-specific interfaces for C or C++; or interfaces based on HTTP and XML. When a third-party system is hosted remotely Business Interlinks framework make invocations in XML format over a secure HTTP channel. In this respect our availability requirement is the same as that for Application Messaging.

The following setup shows a highly available forward proxy architecture for both Application Messaging and Business Interlinks:


DNS
portal.corp.com IN A 123.123.123.100
User
Internet
Firewall
Loadbalancer
Weserver
Webserver
Webserver
Webserver
Appserver
Appserver
Appserver
Appserver
Data
LDAP
LDAP
Forward Proxy 1
10.0.1.51
Port 80/443
Forward Proxy 2
10.0.1.52
Port 80/443
Forward Proxy
Loadbalancer
10.0.1.50
Port 80/443

Chapter 5 Database Server Clustering

PeopleSoft does not provide any support for Database Clustering. It is up to the customers to investigate and select one of the PeopleSoft certified Databases and one of it's (database's) supported clustering mechanisms for their applications. Following is a general overview of the various types of clustering configurations available for a database type that can be used with a PeopleSoft system.

Microsoft SQL Sever Clustering

Microsoft SQL server supports Active/Passive clustering for SQL Server 7 and both Active/Passive and Active/Active clustering on SQL Server 2000. In the active/passive mode one node is primary and the other is a backup, both nodes share a highly available disk array which is dynamically "owned" by the active node in the cluster. In the active/active mode both the nodes are active but they are active on different databases. The first node is primary for say database A and backup for database B. The second node in the cluster is backup for database A and primary for database B. The nodes share two highly available disk arrays, one for database A and the other for database B with "ownership" of each belonging to the respective active unit. In the event of a failure of one of the nodes the backup database instance on the functional node assumes active role. Under these circumstances the functional unit takes ownership for both the disk arrays and runs both database A and B. With Windows 2000 Datacenter it is possible to do a 4 way active/active clustering but the concept of sharing remains the same.

The following link provides a list of documents about database clustering in the Windows NT/2000 environment for SQL Server 7 and 2000.

http://www.sql-server-performance.com/clustering_resources.asp

IBM DB2 Universal Database Clustering

Depending on the HW platform chosen for DB2 UDB there exists different clustering configurations for high availability. In operation the clustering schemes are similar and consists of sharing a highly available disk array among multiple nodes in a cluster. The following link provides information about various DB2 UDB related documents and papers:

http://www-4.ibm.com/software/data/pubs/papers/#dbpapers

HACMP

IBM DB2UDB uses High Availability Clustered Multi-Processing (HACMP) on AIX platform for providing a highly available computing environment. HACMP facilitates the automatic switching of users, applications, and data from one system to another in the cluster after a hardware or software failure. An HACMP cluster is a group of 2 to 32 IBM RS/6000 servers, configured to provide highly available services by sharing highly available disk arrays amongst the cluster nodes. A complete High Availability setup includes many parts, one of which is the HACMP software. Other parts of an HA solution come from AIX and the logical volume manager (LVM). The following document provides instructions to create and test an HACMP configuration:

http://www-4.ibm.com/software/data/pubs/papers/db2eeeaix/db2ee-aixhacmp.pdf

MSCS

IBM DB2UDB uses Microsoft Cluster Server (MSCS) on Windows NT/2000 platform for providing a highly available computing environment. Similar to HACMP a complete High Availability setup includes many parts, one of which is the MSCS software. The following document provides instructions to create and test an MSCS configuration for DB2 UDB:

http://www-4.ibm.com/software/data/pubs/papers/mscseee/mscseee.pdf

SUN Cluster

IBM DB2UDB uses Sun Cluster on Solaris platform for providing a highly available computing environment. Similar to HACMP a complete High Availability setup includes many parts, one of which is the SUN Cluster software. Other parts of an HA solution include a volume manager, which can be either Solstice DiskSuite or VERITAS Volume Manager. The following document provides instructions to create and test a Sun Cluster configuration for DB2 UDB:

http://www-4.ibm.com/software/data/pubs/papers/suncluster/suncluster.pdf

Oracle (OPS/RAC)Clustering

Oracle Parallel Sever (OPS) renamed to Oracle Application Cluster (RAC) for Oracle9i uses the Parallel Fail Safe (PFS) feature to provide a redundant and fault resilient parallel database architecture. Fault resilience is achieved by implementing the ability to recover from (N-1)-node failures in an N-node cluster. In this scheme, as long as one cluster node is available, Oracle Parallel Server can dynamically reconfigure and keep processing transactions for all the databases, albeit at reduced capacity of 1/(N –1). OPS provides both warm and hot standby modes. Documentation on concepts, install and administration of PFS can be obtained from:

http://otn.oracle.com/docs/deploy/availability/content.html

IBM Informix Dynamic Server (IDS) Clustering

IBM Informix has two modes of high availability configuration – High Availability Data Replication (HDR) and Enterprise Replication.

HDR allows a central database server instance to be replicated to a single secondary server. This form of data replication creates a "hot" standby server in case of failure at the primary site. HDR enables the primary and secondary servers to switch roles automatically, without operator intervention. The secondary server can become the primary server in as few as five seconds. The primary and secondary servers can be located anywhere –in the same computer room or geographically separated.

Enterprise replication provides a scalable platform for supporting the high demands of enterprise wide data replication systems. In this scheme an efficient log-based transaction capture and parallel distribution mechanism is implemented, which is integrated with the database architecture. HDR has also used a similar mechanism but restricts the distribution to a backup node only whereas Enterprise Replication implements a bidirectional replication with other active nodes.

Additional information including white paper can be accessed at:
http://www-4.ibm.com/software/data/informix/ids/

Sybase ASE Companion Server Clustering

Sybase Replication Server

Sybase Replication Server uses the Warm Standby feature, where databases are kept in close synch using a database replication mechanism. The primary and secondary servers can be located anywhere –in the same computer room or geographically separated. For High Availability Replication Server 12.1 and above supports the HA Fail-over feature of Sybase ASE 12.5 in a cluster configuration.

The following link provides documents to configure Sybase Replication Servers:

http://www.sybase.com/products/eaimiddleware/replicationserver

Sybase Adaptive Server Enterprise (ASE)

Sybase Adaptive Server Enterprise (ASE) uses Companion Server option to integrate with third party Hardware and Software High Availability Solutions. This option allows to configure 2 Adaptive Server Enterprise servers as companions in either asymmetric (master-slave) or symmetric companion

(active/active hot standby) configuration to create a hot standby capability.
Companion Server option works with the following high availability solutions.

- Sun Microsystems – Sun Cluster
- IBM (AIX) – HACMP
- Hewlett-Packard – ServiceGuard
- Compaq – TruCluster
- Microsoft – Windows NT MSCS

Appendix A – Special Notices

All material contained in this documentation is proprietary and confidential to Oracle Corporation is protected by copyright laws, and subject to the nondisclosure provisions of the applicable Oracle agreement. No part of this documentation may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, including, but not limited to, electronic, graphic, mechanical, photocopying, recording, or otherwise without the prior written permission of Oracle Corporation.

This documentation is subject to change without notice, and Oracle Corporation does not warrant that the material contained in this documentation is free of errors. Any errors found in this document should be reported to Oracle Corporation in writing.

The copyrighted software that accompanies this documentation is licensed for use only in strict accordance with the applicable license agreement, which should be read carefully as it governs the terms of use of the software and this documentation, including the disclosure thereof. See Customer Connection or PeopleBooks for more information about what publications are considered to be product documentation.

Oracle, PeopleSoft, the PeopleSoft logo, PeopleTools, PS/nVision, PeopleCode, PeopleBooks, and Vantive are registered trademarks, and *PeopleTalk* and "People power the internet." are trademarks of Oracle Corporation. All other company and product names may be trademarks of their respective owners. The information contained herein is subject to change without notice.

Information in this book was developed in conjunction with use of the product specified, and is limited in application to those specific hardware and software products and levels.

Oracle may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents.

The information contained in this document has not been submitted to any formal Oracle’s PeopleSoft test and is distributed AS IS. 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’s PeopleSoft 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. Any pointers in this publication to external Web sites are provided for convenience only and do not in any manner serve as an endorsement of these Web sites.

Appendix B – Validation and Feedback

This section documents that real-world validation that this Red Paper has received.

Customer Validation

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

Field Validation

PeopleSoft is working with PeopleSoft Global Services to get feedback and validation on this document. Lessons learned from these field experiences will be posted here.

Appendix C – Revision History

Authors

Sheshi Sankineni

Lakshmi Gourabathina

Hemanth Sundaram

Simon Sy

Susan Chen

Reviewers

The following people reviewed this Red Paper:

- Richard Sze, Director of PeopleTools Server Tools
- Ryan McAfee, VP of PeopleTools Strategy
- Kirk Chan, PeopleTools Strategy
- Edgar Vasquez, PeopleSoft Consulting
- Michael Hillerman, PeopleTools Strategy
- Chris Heller, PeopleTools Strategy

Revision History

1. 6/26/2002: Version 1 updated for 8.4 from V2 of 8.1 document.
2. 8/23/2002: Updated WebLogic proxy port specification for IIS, iPlanet and Apache.
3. 11/5/2002: Added workarounds for WebLogic clustering and updated for 8.42.
4. 3/11/2003: Fixed UNIX script instructions to include .sh.
5. 9/24/2003: Updated server ID and clone ID section for WebSphere
6. 3/25/2004: Version 2 updated for 8.44
7. 11/15/2005: Updated for OAS Clustering for 8.47.
8. 12/3/2007: Updated Clustering setup for PT 8.49

9. 1/29/2009: Minor text formatting updates to WebSphere 6.1 Clustering steps for PT 8.49

10. 1/29/2009: Added WebSphere 7.0 Clustering steps for PT 8.50
11. 10/7/2009: Made updates to Weblogic Clustering steps for PT 8.50

White Paper Title
November 2007
Author: [OPTIONAL]
Contributing Authors: [OPTIONAL]

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