

An Oracle White Paper
September 2009

Installing AIA on an Oracle SOA Suite Cluster Deployment

Introduction.....	1
Scope of this document	3
Software pre-requisites.....	5
Document References	5
Topologies considered in this document.....	6
Topology A	6
Topology B	8
Oracle Service Registry Topology Considerations	9
Ensuring that the SOA is setup correctly.....	12
Setting SOA parameters	12
Verifying SOA Cluster setup	14
Preparing the APPHOSTn and WEBHOSTn for AIA deployment.....	16
Setting up APPHOSTn.....	17
Setting up WEBHOSTn.....	19
Steps for Deployment of AIA FP to the SOA cluster.....	20
On APPHOST2.....	20
On WEBHOSTn (Applicable only to Topology A).....	21
On APPHOST1	22
Post Install actions for AIA FP installation on SOA cluster	23
Steps for Deployment of AIA PIPS on AIA FP clustered deployment.....	27
On APPHOST2.....	27
On WEBHOSTn (Applicable only to Topology A).....	29
On APPHOST1	29
Post Install actions for PIP deployments on AIA FP cluster.....	32
Upgrading AIA installations on SOA cluster	33
Conclusion.....	34

Introduction

Oracle Application Integration Architecture (AIA) Foundation Pack is a comprehensive SOA based integration product from Oracle that not only provides an architecture blue print for true SOA based integration of enterprise applications, but also provides pre-built standards-based content required for integration and Programming Models to achieve superior performance, scalability, extensibility, security etc. In addition, AIA Foundation Pack (AIA FP) contains inbuilt tools for SOA governance, Integration validation, error handling etc making it a comprehensive Integration Product for enterprises to jump start on Integration initiatives. AIA FP runs on Oracle Fusion Middleware (FMW) SOA Suite.

AIA Process Integration Packs (AIA PIPs) are pre-built end-to-end integrations built on AIA Foundation Pack. PIPs are products that implement a specific set of business process between a specific set of Applications (Oracle E-Business Suite, Siebel, SAP etc). Besides providing immediate functional value, PIPs derive all the principles and infrastructure from FP, there by delivering a complete SOA implementation along with superior ownership experience.

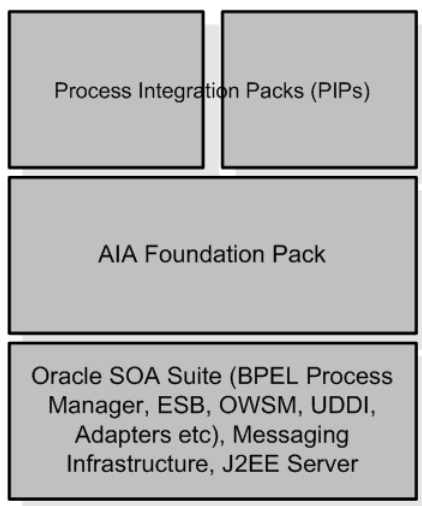


Figure1: AIA products and their relationships

AIA FP and PIPs are installable products that can be installed against installations of SOA Suite, Database, OSR and participating Applications (eg. Siebel, PeopleSoft). AIA Installation/deployment ensures all AIA pre-packaged content is delivered at the end of

the installation along with establishing connectivity and configuration, so that FP and PIPs are ready to use.

Deploying AIA to a highly available enterprise deployment provides the following distinct advantages.

- Because AIA integrates business-critical Enterprise software, it is important to ensure that there is no single point of failure and that system maintenance and failure recovery can happen without impacting the business.
- Workload can be distributed effectively to multiple nodes allowing for optimal usage of resources.
- New resources can be added or removed easily. For example, an Order to Cash Integration could need more resources during a specific season of the year
- Specific process flows can be provisioned with more resources than other flows based on the kind of operation they perform and the frequency of usage.
- High availability is ensured at all levels (web tier, application tier and database tier). Application and data tiers are also secured as they lie within firewalls.

This paper will give step-by-step instructions on installing AIA on a SOA Cluster along with the explanation of the overall procedure, so that the same instructions can be used to extrapolate the content of this paper to suit your own Topology.

AIA FP runs completely on Oracle SOA Suite and the underlying FMW infrastructure. For this reason, AIA FP High Availability is directly derived from the underlying enterprise deployment topology implemented for SOA Suite and the data tier

Scope of this document

As mentioned earlier, AIA FP runs completely on Oracle SOA Suite and the underlying FMW infrastructure. For this reason, AIA FP High Availability is directly derived from the underlying enterprise deployment topology implemented for SOA Suite and the data tier. Please refer Oracle Application Server 10gR3 Enterprise Deployment Guide to learn more.

This whitepaper contains two major sections. The first one explains the instructions to install AIA Foundation Pack on a SOA cluster and the subsequent section explains how to install any PIP on top of an FP cluster. Lastly there is a section on how to upgrade a given AIA cluster install to the latest release of AIA.

This whitepaper applies to SOA Suite 10.1.3.4 and above running on Oracle Application Server 10gR3 (OC4J). The applicable AIA version is the latest generally available version of AIA

Both AIA FP and AIA PIPs are composed of several BPEL/ESB processes, configurations, WSDLs/XSDs, queues, database objects, J2EE applications etc. all of which are artifacts created or configured on FMW.

Since these artifacts are large in number and have dependencies, for single node installations, AIA provides a wizard-based Installer that collects the software information and kicks off several install scripts and tools to deploy and configure all these artifacts automatically, thereby providing an out-of-the-box experience for customers.

However HA topologies can widely vary depending on the requirement and so the steps to deploy AIA FP to Highly Available topologies, the underlying install scripts need to be executed manually based on the desired topology.

There are some special considerations when deploying AIA to the SOA HA deployment. These are explained in the section "Topologies considered in this paper"

This paper provides step-by-step instructions to deploy AIA FP and any SOA Suite based PIP on a given HA deployment of SOA Suite. When providing the steps we will constantly refer to the topology and AIA considerations sections there by allowing users to extrapolate the instructions if they have a different topology.

Software pre-requisites

The following are the basic Software applications required for an AIA Installation.

- Oracle SOA Suite10gR3 (10.1.3.4) – Oracle SOA Suite, J2EE server, HTTP Server
- Oracle Database (10.2.0.1) – Refer Application Server certification matrix to know all supported database versions
- Oracle Service Registry 10.3.0.0 - Optional

The above information is dependent on the version on AIA Foundation Pack that is being installed. Refer AIA Installation Guide on Oracle Metalink to know the exact software requirements

Please review Chapter 1 and Chapter 2 of the AIA Installation Guide to understand the basics of AIA Installation.

It is required that you have set up the SOA Suite Cluster as per the Enterprise Deployment Guide before attempting the steps provided in this paper. Alternatively you can set up a simpler topology as mentioned in the Topology section of this paper.

Document References

Oracle AIA Installation Guide: Available on Oracle My Oracle Support (Metalink)

Oracle Application Server 10gR3: Enterprise Deployment Guide -
http://download.oracle.com/docs/cd/E10291_01/core.1013/e10294.pdf

This paper does not address clustering at data-tier, Security considerations, disaster recovery etc. This is because there are no AIA considerations or special recommendations required. More information on these topics can be found at the respective product documentation locations:

- Security in Clusters: http://download.oracle.com/docs/html/A96687_01/manage.htm
- Database clustering:
<http://www.oracle.com/technology/products/database/clustering/index.html>
- Disaster Recovery:
http://download.oracle.com/docs/cd/B14100_14/install.1012/install/ha_dr.htm

Topologies considered in this document

The Enterprise Deployment Guide (EDG) provides a reference topology that ensures that the SOA deployment is highly available at all tiers (Web tier, Application Server Tier, Data tier and Security tier).

As the EDG rightly points out there are several variations possible to the proposed topology depending upon user requirement.

For the purpose of this paper, we would provide 2 variations

Topology A – High Availability at all tiers (reference topology used in EDG)

Topology B – High Availability only at the Application tier

Installation steps would be almost similar for both topologies and wherever there is variance it will be highlighted.

For any other topology apart from the above, you can easily extrapolate the required steps based on the explanation.

Topology A

In this topology the load balancer accepts external requests and in turn redirects requests to Oracle HTTP servers (Apache Servers). These HTTP servers in turn point to one of the Application Server nodes that host BPEL/ESB and other J2EE applications. The Application Server nodes in turn point to an Oracle RAC database.

Please refer figure 2, to see a complete representation of this topology. In addition, the figure also provides summary information of different AIA artifacts to be installed at each tier

- **Load Balancer** – Balances external requests to one of the Oracle HTTP Servers (WEBHOSTn). There would be two virtual hosts one for external requests that use HTTPS: 443 called the soa.mycompany.com and an internal virtual host soainternal.mycompany.com

Note: If you do not have HTTPS requirement to allow external access to your system, you would only use the internal virtual host

- **WEBHOST1/WEBHOST2** – Servers that run Oracle HTTP servers in their Oracle Homes. Requests are redirected to the APPHOSTn

This is where AIA Enterprise object library artifacts like EBO, EBS etc (which are XSDs and WSDLs) reside.

- **APPHOST1/APPHOST2** - Servers that host Oracle SOA Suite Applications and the underlying J2EE server infrastructure.

In this paper we will consider only BPEL and ESB applications of SOA Suite. For other applications such as OWSM, there are no AIA artifacts to be deployed. So please refer the Enterprise Deployment Guide directly and follow SOA Suite best practices.

This tier hosts AIA delivered BPEL/ESB processes, J2EE applications, datasource configurations etc. AIA Home will be made available in each host. AIA Home is more of a design time repository of contents that are deployed to the OracleAS Oracle Home (called SOA HOME here). There are some run-time dependencies with AIA Home as well, making it necessary to be on each server.

- **DBHOST1/DBHOST2** – Servers that host the instances of the RAC database. This is used as a dehydration store for SOA. The same database can also store AIA specific data like XREF, CAVS/BSR data, AIA Queues etc.

OID is not considered in this topology, as there are no AIA specific considerations. Please follow the SOA Suite best practices.

- **DBHOST3/DBHOST4** – It is not necessary to use the SOA database for AIA schema/data also. Users can optionally choose to use another database to store AIA specific data like XREF, CAVS/BSR data, AIA Queues etc.

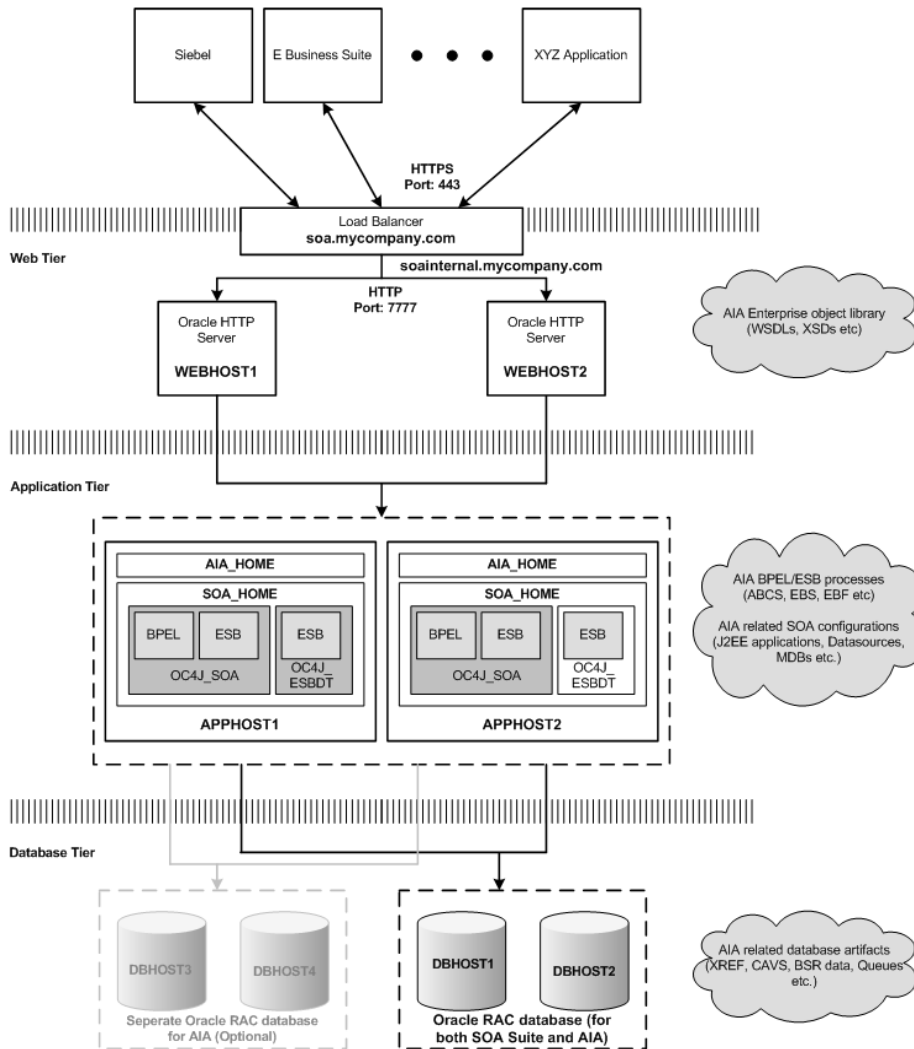


Figure 2: Topology A for AIA Foundation Pack deployment

Topology B

In this topology the load balancer accepts external requests and in turn redirects requests to servers that host both Oracle HTTP Server and BPEL/ESB applications in the same Oracle Home. The Application Server nodes in turn points to a single Oracle database that stores both AIA and SOA data. The Load balancer has one virtual host to redirect incoming requests to one of the two servers

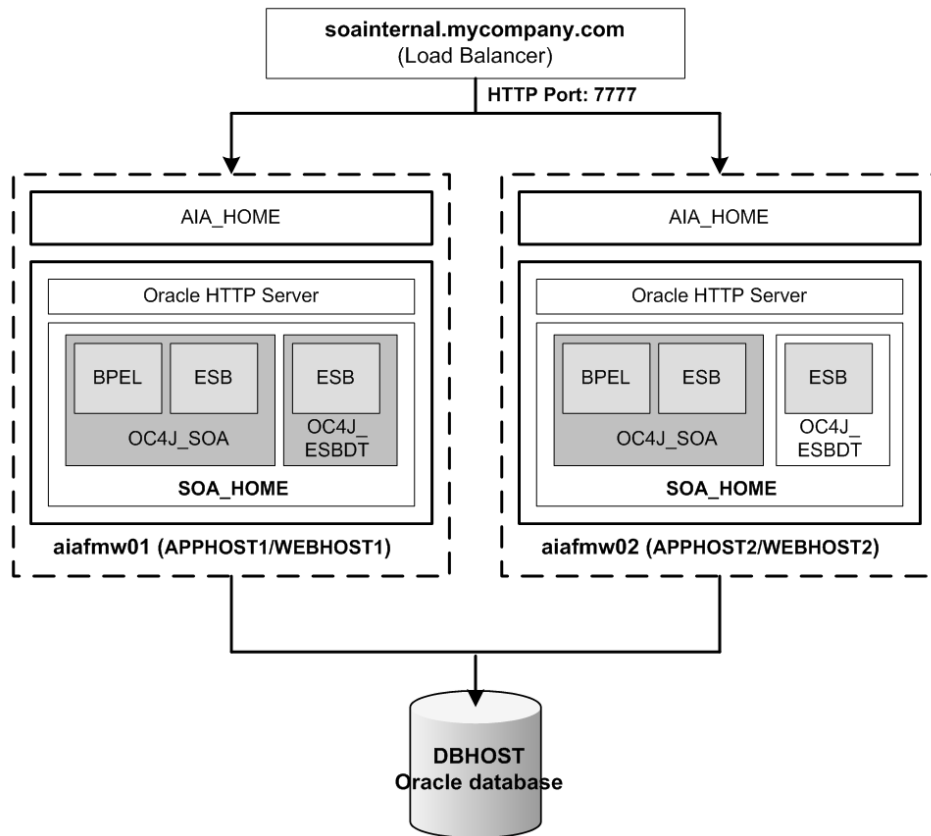


Figure 3: Topology B for AIA Foundation Pack deployment

Oracle Service Registry Topology Considerations

As a part of AIA Installation AIA services (EBO, EBS, ABCS etc) are published to BSR (Business Service Repository). Details of OSR (Oracle Service Registry) is collected during installation, and AIA services have to be manually published to OSR as a post install activity as detailed in the AIA Installation Guide.

BSR is an AIA delivered repository and is AIA's key value proposition for SOA governance. BSR provides both design time and run time information about services, and models their interdependencies to represent AIA process flows and PIPs.

BSR infrastructure is composed of J2EE applications and Database artifacts. AIA Foundation Pack's EBO/EBS are represented in this infrastructure and PIPs will use BSR to publish metadata about PIP specific services and their inter-dependencies.

Installation of BSR infrastructure is mandatory and is automatically installed with AIA Foundation Pack installation.

Installation of BSR infrastructure is mandatory and is automatically installed with AIA Foundation Pack installation. OSR on the other hand is a separate Oracle product that is used to publish run-time information about raw services that make up AIA.

OSR on the other hand is a separate Oracle product that is used to publish run-time information about raw services that make up AIA. Services thus published can be discovered across the enterprise. OSR will contain information of all EBS and ABCS service implementations.

OSR is deployed as an application in oracle SOA Suite. There are three options AIA installation provides even in a single server installation.

- **No OSR Installation** – AIA has no run-time dependency on OSR

Hence it is possible to install and run AIA without having an OSR installation. AIA Installer provides the option to skip the screen where OSR details are collected.

Note: At any later point, if a decision to include OSR is made, AIA services can be published to that OSR. Additional Information is available in the AIA Core Infrastructure Components Guide.

- **OSR can be installed in the same Application Server** that hosts AIA BPEL/ESB processes

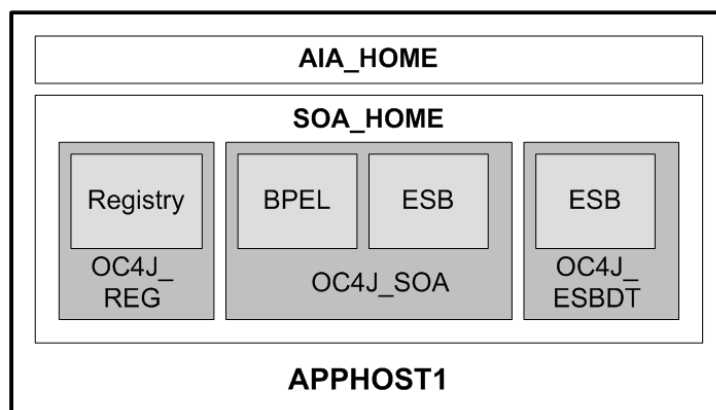


Figure 4: OSR in the same APPHOST as AIA

- **OSR can be installed in a completely different Application server.** This is important when AIA services are to be published in an enterprise-wide service registry

that is also being used to publish non-AIA services. (Publishing AIA services to remote servers is supported from FP release 2.2.1 onwards)

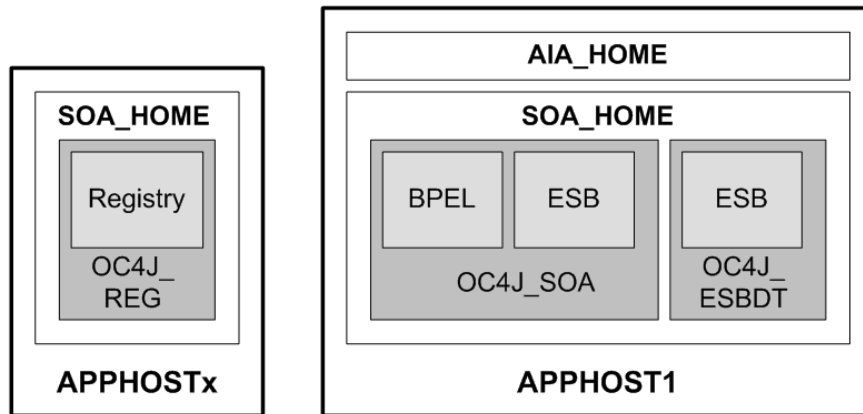


Figure 5: OSR in a different APPHOST than AIA

In addition, Oracle Service Registry product also provides ways to achieve clustered OSR implementations thereby providing high availability for registry access. This again is optional.

More information is available at

http://www.oracle.com/technology/tech/soa/uddi/osr_cluster_config.pdf

Ensuring that the SOA is setup correctly

Since AIA is an application running completely on SOA Suite it is important to ensure that SOA setup was performed correctly, before proceeding with AIA Installation

Setting SOA parameters

The following are SOA Suite parameters to be set up correctly to ensure a smooth AIA installation.

1. Memory Settings

Login to Oracle Enterprise Manager. Perform the below steps on both APPHOSTs

- Click oc4j_soa".
- Click Administration tab.
- Click Server Properties and adjust settings on this page as follows:
 - Initial Heap Size: Set to 1024M
 - Max Heap Size: Set to 1024M
- Under Options, ensure that the following parameter is updated or set:
 - -XX:MaxPermSize=512M
 - -XX:MaxNewSize=614m
 - -XX:NewSize=614m
 - -XX:AppendRatio=3
 - -XX:SurvivorRatio=6

Note: These are recommended values. Actual values that you set depend on your system configurations. When running on dedicated hosts, you can set the JVM heap size as high as possible. However this number is constrained by the operating system's addressable memory space.

2. Order of Restart

Important: It is not necessary to perform restart right now. Whenever SOA Servers have to stopped or started in the later sections of this paper , ensure that you follow the method below.

Always restart containers separately in the following manner

On APPHOST1

```
./opmnctl shutdown (this is to stop all containers)

./opmnctl start (this starts the opmn)
./opmnctl startproc process-type=HTTP_Server
./opmnctl startproc process-type=ADMIN (or whichever is your admin node name)
./opmnctl startproc process-type=oc4j_esbdt (or whichever is the name of the ESB DT container)
./opmnctl startproc process-type= oc4j_soa (or whichever is the name of the BPEL and ESBRT container)
```

On APPHOST2,

```
./opmnctl shutdown (this is to stop all containers)

./opmnctl start (this starts the opmn)
./opmnctl startproc process-type=HTTP_Server
./opmnctl startproc process-type= oc4j_soa (or whichever is the name of the BPEL and ESBRT container)
```

To know the status of the cluster execute the following from the node that is alive

From the folder <soahome> /opmn/bin execute the following command to see the status (linux)

```
./opmnctl @cluster status
```

3. Open the server.xml from ORACLE_HOME /j2ee/oc4j_soa/config and make sure that the lines beginning with following are in the given order.

```
<application name="orabpel" ...>
<application name="esb-rt" ...>
```

If BPEL+ ESBRT is installed in a different container other than oc4j_soa, the path to server.xml reflects that container name instead of oc4j_soa. Perform the above changes in both APPHOSTs

4. Open the file SOA_HOME /integration/esb/config/esb_config.ini

Add the following lines at the end of the file (This will ensure that the ESB DT starts up correctly)

```
PingInterval=30
PingCount=30
```

Perform the above changes in both APPHOSTs. Restart SOA using the commands in step 2.

Verifying SOA Cluster setup

The following are basic verification steps (not comprehensive) to ensure that your SOA cluster is setup correctly as per the EDG.

- Navigating to the BPEL, ESB consoles successfully using the external Load Balancer URL as well as each of the internal virtual host.
 - Ensuring that the EM console shows all participating nodes in active state. There must be only one Active Design time.
 - Ensure that ESB Design-time is completely up and running by accessing the following URL
 - <http://host:port/esb/dtStatus.jsp>. This page should return a true
 - Deploying a sample BPEL process (shipped with BPEL) and a sample ESB process (again shipped with ESB) and ensuring that it is successfully deployed on the SOA cluster (available on both APPHOST1 and APPHOST2)
 - Create a sample system from ESB Console.
 - After you complete, remember to undeploy the samples and delete the previously created system.
 - Verify that you don't have any existing applications in your SOA cluster starting with "default_"
 - For example if you had deploy the LoanFlow demo, it would've created such applications. To check this
 - Open the EM Console
 - Navigate to the container
 - In the Applications tab, drill down from BPEL>orabpel.
 - Select any application starting with "default_" and undeploy
 - Use HTTP monitoring to ensure that both HTTP Servers serve requests
 - Accessing RAC database using a tool like TOAD to perform successful query, insert operations (using individual DBHOST TNS entries as well as RAC TNS entry)
 - Ensure that all other configuration (like updating ESB metadata, Configuring slide repository etc) is performed correctly as per the Enterprise Deployment Guide.
- As per the Enterprise deployment guide ensure that the soapServerURL is set to internal virtual host <http://soainternal.mycompany.com> and the soapCallbackUrl to external virtual host <http://soa.mycompany.com>

If you are not using SSL for external access, both these properties should point to <http://soainternal.mycompany.com>

Ensure that the cluster flag is set to true

soapCallbackUrl	BPEL soap callback URL	http://soa.mycompany	<p>This URL is sent by the server as part of the asynchronous callback address to the invoker.</p> <p>The hostname and port for this URL should be customized to match the hostname of your system and the port of your HTTP gateway.</p>
soapServerUrl	BPEL soap server URL	http://soainternal.myc	<p>This URL is published as part of the SOAP address of a process in the WSDL file.</p> <p>The hostname and port for this URL should be customized to match the hostname of your system and the port of your HTTP gateway.</p>

Preparing the APPHOSTn and WEBHOSTn for AIA deployment

At the end of this section, you will make available all the necessary AIA software in the APPHOSTn and setup the configuration required to perform the manual deployment detailed in the next section (using the install scripts delivered through this section)

To begin with, the SOA cluster that you setup should have APPHOST1, 2 and WEBHOST1, 2 (applicable in the case of Topology A only) up and running. Your ESB DT container should be down on one APPHOST and should be alive in the other APPHOST. Refer section "Ensuring that the SOA is setup correctly" for the correct steps to query status of the SOA Server.

```
[oracle@aiaafmw01 bin]$ opmnctl @cluster status
```

Processes in Instance: APPHOST2.APPHOST2			
ias-component	process-type	pid	status
OC4JGroup:esbdt_group	OC4J:oc4j_esbdt	N/A	Down
OC4JGroup:soa_group	OC4J:oc4j_soa	25930	Alive
OC4JGroup:default_group	OC4J:home	25928	Alive

Processes in Instance: APPHOST1.APPHOST1			
ias-component	process-type	pid	status
OC4JGroup:esbdt_group	OC4J:oc4j_esbdt	23630	Alive
OC4JGroup:soa_group	OC4J:oc4j_soa	15893	Alive
OC4JGroup:default_group	OC4J:home	18406	Alive

Processes in Instance: WEBHOST1.WEBHOST1			
ias-component	process-type	pid	status
HTTP_Server	HTTP_Server	25927	Alive

Processes in Instance: WEBHOST2.WEBHOST2			
ias-component	process-type	pid	status
HTTP_Server	HTTP_Server	18405	Alive

Figure 6: opmn status for Topology A

```
[oracle@aiafmw01 bin]$ opmnctl @cluster status
```

Processes in Instance: APPHOST2.aiafmw02			
ias-component	process-type	pid	status
OC4JGroup:esbdt_group	OC4J:oc4j_esbdt	N/A	Down
OC4JGroup:soa_group	OC4J:oc4j_soa	25930	Alive
OC4JGroup:default_group	OC4J:home	25928	Alive
HTTP_Server	HTTP_Server	25927	Alive

Processes in Instance: APPHOST1.aiafmw01			
ias-component	process-type	pid	status
OC4JGroup:esbdt_group	OC4J:oc4j_esbdt	23630	Alive
OC4JGroup:soa_group	OC4J:oc4j_soa	15893	Alive
OC4JGroup:default_group	OC4J:home	18406	Alive
HTTP_Server	HTTP_Server	18405	Alive

Figure 7: opmn status for Topology B

Setting up APPHOSTn

Important: The following steps have to be performed in both APPHOST1 and APPHOST2

1. Launch the AIA installer as per instructions in the AIA Installation Guide. Provide the AIA Home directory and choose the option "Copy AIA Software Only" option. Choosing this option and clicking 'Next' and 'Install', should directly copy only the content to the selected AIA Home without prompting for SOA Server, Database or Service Registry information.
2. The previous step will deliver AIA_HOME (containing all FP content and scripts) on APPHOSTn.
3. Copy the deploy.properties.tmpl file template from AIA_HOME /Infrastructure/install/templates folder to the location, AIA_HOME /config and rename it to deploy.properties
4. In this deploy.properties file,
 - For the http.host/http.port provide the internal virtual host details (soainternal.mycompany.com)
 - For the j2ee.hostname provide the information of APPHOST1/APPHOST2
 - For the database provide the details of the Database that you want to host AIA artifacts. This can be the same as SOA database or a separate database as mentioned in the Topologies Section of this paper.

If you are using RAC provide the information of one of the instances that is a part of the RAC. For example in deploy.properties of APPHOST1 use

DBHOST1 SID and for APPHOST2 use DBHOST2. Later we will modify references of data-sources and connection pools to point to RAC.

If you are using XA and Distributed Transaction Processing (DTP) provide the corresponding DTP service name for each APPHOST

For more information on DTP and XA for SOA Suite, refer

<http://www.oracle.com/technology/tech/soa/pdf/oracle-soa-suite-xa-rac-guide.pdf>

- The port information should be available from the runtime port information on em console
 - Use "oc4j_soa" for the instance name. (Note: This must be the instance running BPEL)
 - For the service registry details, you can provide the information of the local node or a remote server or skip the fields. Refer Oracle Service Registry Topology Considerations for more information. If you want to completely skip publishing AIA artifacts to service registry leave these fields empty
5. In the ant-orabpel.properties file (found in the directory <SOA_HOME> /bpel/utilities) ensure that:
- hostname= soainternal.mycompany.com
 - http.port=7777 (internal LBR port)
 - cluster= false
 - oc4jinstancename = the OC4J instance where BPEL is installed. Example oc4j_soa for this topology. This should NOT be the soa_group OC4J container of the cluster
 - j2ee.hostname= The host name of APPHOST1/APPHOST2
 - asinstancename= (this setting should not have a value. It should be blank.)
6. Complete the following steps.
- Open the file aiaenv.sh (aiaenv.bat in the case of Windows) under AIA_HOME /bin.
 - Set the values for all properties like aia_home, ant_home, oracle_home, java_home, oracle_registry_home, aia_instance (<oracle_home> /j2ee/<container where BPEL is located> /applications)
 - Execute the following command

```
source aiaenv.sh (linux)
```

```
aiaenv.bat (windows)
```

- Execute the following command

```
cd ANT_HOME/bin
chmod 775
```

Setting up WEBHOSTn

The following steps have to be done on all webhosts.

1. Enable directory browsing in the Apache server of both WEBHOST1 and WEBHOST2 by editing the httpd.conf file. This file is at <SOA_HOME>/Apache/Apache/conf/httpd.conf.
2. To edit the httpd.conf file:
 - Open the file using a text editor, such as vi.
 - Search for Options FollowSymLinks MultiViews. There are two occurrences.
 - Add the word Indexes to the end of Options FollowSymLinks MultiViews for the second occurrence.
 - After adding it, the text looks like this: "Options FollowSymLinks MultiViews Indexes"
3. Restart HTTP Apache server. Important: The following steps have to be performed in both APPHOST1 and APPHOST2

Note: if you prefer to have your web tier lightweight and don't want APPHOSTs to access your web tier across the DMZ to retrieve the content, you can later bundle up all the AIA content as separate .ear file and deploy it to your APPHOSTn and set redirections accordingly. This paper will not elaborate further on this approach and will consider the content to be in WEBHOST.

In case of Topology B, there are no separate WEBHOSTs. So follow the same steps above in APPHOST1 and APPHOST2

Steps for Deployment of AIA FP to the SOA cluster

The following is the high level steps that we will be performing to complete the deployment of AIA FP

We would execute a few install scripts to complete the installation on each node.

- On APPHOST2 we would execute the install scripts that correspond to steps A, B and E of section 4.
- On WEBHOST2 and WEBHOST1 we would copy the required HTTP content into these hosts, covering step B of section 4 in each node.

This step is required only in the case of Topology A

- On APPHOST1, we would again copy all the AIA content and the install scripts to the disk. We would use the AIA Installer to cover all steps (A, B, C, D and E) mentioned in section 4.

Important: You need to follow the above actions in the same sequence

On APPHOST2

The following steps have to be followed on APPHOST2

1. Shutdown SOA Server on APPHOST1. Refer section “Ensuring that the SOA is setup correctly” for the correct steps to perform this.
2. Ensure that APPHOST2 and WEBHOST1,2 (in the case of Topology A only) are up and running.
3. Execute the following scripts one after the other, while being in the “AIA_HOME/Infrastructure/install/scripts” directory.

```
ant --noconfig -buildfile FPCommonConfig.xml replaceFPInstallProperties
ant --noconfig -buildfile FPConfiguration.xml CommonConfigurations
```

4. This will complete setting up the required files and AIA XPATH patches in SOA_HOME as well as copy the web content to Apache/htdocs. This completes step A and B of section 4.

In the case of Topology A, we will later copy this AIA content to the WEBHOSTn as the web server really resides there.

5. Next we need to create/deploy Datasources, Connection pools, AIA j2ee application/servlet and Errorhandling MDB.
6. Execute the following scripts one after the other, while being in the "AIA_Home/Infrastructure/install/scripts" directory

```
Ant --noconfig -buildfile FPCCommonConfig.xml CreateCommonDatasources
ant --noconfig -buildfile FPCAVSInstall.xml createSchedulerDataSource
ant --noconfig -buildfile FPErrorHandlingInstall.xml DeployESBErrorListenerMDB
ant --noconfig -buildfile FPCAVSInstall.xml DeployAIAUI
```

7. You should see connection Pools, data sources, MDB etc
8. This completes step E of section 4.
9. Now when you check the EM console, no configurations (connection pools, datasources, MDB etc.) should have been created for APPHOST1

JDBC Resources						
Application: All						Page Refreshed Feb 10, 2009 4:42:46 AM PST
Data Sources						
Create						
Attributes						
Name	Application	JNDI Location	Connection Pool	Managed by OC4J	Test Connection	Delete
"AIADataSource"	default	jdbc/AIADDataSource	"AIAConnectionPool"	✓		

10. In case you see any of them, manually undeploy them using the undeploy actions on EM.
11. Now bring down SOA server on APPHOST2

On WEBHOSTn (Applicable only to Topology A)

The steps below are only applicable to Topology A. If you are using Topology B, skip this section.

We will copy the HTTP content created in the Apache folder on APPHOST2, to the real Apache folders on WEBHOST1 and WEBHOST2.

1. From the directory <SOA_HOME> /Apache/Apache/htdocs on APPHOST2, copy "recursively" (include all subfolders and their contents) to the same directory structure under WEBHOST1. If asked to overwrite any files choose the option to overwrite.

2. Similarly, from APPHOST2, copy the contents from the directory <SOA_HOME> /Apache/Apache/htdocs “recursively” (include all subfolders and their contents) to the same directory structure under WEBHOST2. If asked to overwrite any files choose the option to overwrite.
3. Restart the HTTP server of both WEBHOST1 and WEBHOST2

On APPHOST1

The following steps have to be followed on APPHOST1

1. Ensure that APPHOST2 is down.
2. Also ensure that APPHOST1 and WEBHOST1, 2 (in the case of Topology A only) is up and running.
3. In APPHOST1 create an empty file called httpd.conf in the folder <SOA_HOME> /Apache/Apache/htdocs

This step is required only in the case of Topology A

4. From AIA HOME directory on APPHOST1, navigate to the folder Infrastructure/install/scripts
5. From the command line execute the command


```
source <AIA_HOME>/bin/aiaenv.sh
ant --noconfig -buildfile FPInstall.xml > FPInstall.log
```
6. Verify if all AIA processes were deployed as per AIA installation guide verification chapters. Verify if AIA Connection pools and AIA data sources are created.
7. Now bring up APPHOST2. This should propagate all BPEL process to APPHOST2.
8. Verify that the AIA datasources and AIA connectionpool exist on APPHOST2.

This completes the deployment of AIA FP on a SOA cluster.

Complete the steps provided in the section below, Post Install Actions for PIP deployments on AIA FP cluster

Post Install actions for AIA FP installation on SOA cluster

The following are the post install actions to be performed after the installation of AIA FP on SOA Cluster. These steps have to be performed for AIA to work correctly and are required to be done before PIPs are installed on the cluster.

1. In APPHOST2's EM console, if you drill down into the oc4j_soa container, you would NOT find the Taskforms deployed.

2. Now on APPHOST2, execute the following command from AIA_HOME/bin directory

```
source aiaenv.sh
```

3. Open the AIAErrorTaskAdministrationProcess folder under AIA_HOME/Infrastructure/ErrorHandling folder.

4. Execute from the same folder

```
ant -noconfig deploytaskform
```

5. This should deploy two task form processes on APPHOST2.

6. On APPHOST1, navigate to the AIAReadJMSNotification process folder under AIA_HOME/Infrastructure/ErrorHandling

7. In the bpel folder open the consumeFaultMsg.wsdl, add the following tag after DestinationName="java:comp/resource/AIA_ERROR_JMS/Topics/AIA.AIA_ERROR_TOPIC"

```
DurableSubscriber="AIAReadJMSNotificationAgent"
```

8. Open the file, AIA_HOME/Infrastructure/ErrorHandling/AIAReadJMSNotificationProcess/bpel/bpel.xml. Add the bolded line in the activationAgents existing snippet. The string for clusterGroupId can be anything

```

<activationAgents>
  <activationAgent className="oracle.tip.adapter.fw.agent.jca.JCAActivationAgent"
partnerLink="consumeFaultMsg">
    <property name="portType">Consume_Message_ptt</property>
    <property name="clusterGroupid">MyCluster</property>
  </activationAgent>
</activationAgents>

```

9. Repeat last three steps on APPHOST2
10. Open the ESB Console and verify the details of each of the systems (AIASystem, DefaultSystem and BPELSystem).

The Values must be as follows. If not, please change as follows and press 'Apply'

Cluster Name: The value defined for Cluster Name in
<SOA_HOME> /integration/esb/config/esb_config.ini or the value in
<SOA_HOME> /j2ee/<container> /applications/esb-rt/META-INF /orion-
application.xml

Virtual Host: Load Balancer URL

Port: Load Balancer Port number

Topic Location and Connection Factory Location: The value for JMS Topic should be the same as the one that is set as DEFERRED_TOPIC in
ESB_PARAMETER table 'DT_OC4J_HOST'.

Normally, these values would be ESBTopics/Topics/ESB_JAVA_DEFERRED
and the factory is OracleOJMS/XATCF

Number of Listeners: The value would be 1

11. If you were using RAC database, then you need to ensure that AIA related
datasources and connection pools point to the RAC URL

On the EM console, navigate to the JDBC resources of the oc4j_soa container and
modify AIAConnection pool information as follows.

Change the JDBC URL to RAC format. Typically this would be of the format

```

Jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=on) (ADDRESS=(PROTOCOL=TCP)(HOST=host1)
(PORT=1521)) (ADDRESS=(PROTOCOL=TCP)(HOST=host2)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=service_name)))

```

Perform this on both APPHOSTs

Additionally, in the AIAConfigurationProperties.xml file) found in
AIA_HOME /config) make sure that the EHAQ.DB.XX details are set as the
individual nodes of the RAC.

So in APPHOST1 set it to one node of the RAC. In the same file in APPHOST2 set the properties to the other node of the RAC.

If you are using XA and Distributed Transaction Processing (DTP) provide the corresponding DTP service name for each APPHOST in the AIAConfigurationProperties.xml file. Also in the Connectionpool on each APPHOST provide the corresponding DTP service name instead of modifying to the RAC URL above.

For more information on DTP and XA for SOA Suite, refer <http://www.oracle.com/technology/tech/soa/pdf/oracle-soa-suite-xa-rac-guide.pdf>

12. In both APPHOST1 and APPHOST2, verify the following values
 - In the ant-orabpel.properties file (found in the directory <SOA_HOME> /bpel/utilities) ensure that:
 - hostname=soainternal.mycompany.com
 - http.port=7777 (internal LBR port)
 - cluster= false
 - local.oc4jinstancename= the OC4J instance where BPEL is installed. Example oc4j_soa for this topology. This should NOT be the soa_group OC4J container of the cluster

Note: This property should be manually added

 - oc4jinstancename = This should be the soa_group OC4J container of the cluster
 - j2ee.hostname= The host name of APPHOST1 OR APPHOST2
 - asinstancename= (this setting should not have a value. It should be blank.)
 - In deploy.properties file (found in the directory <AIA_HOME> /config) ensure that:
 - hostname=soainternal.mycompany.com
 - http.port=7777 (internal LBR port)
 - j2ee.hostname= APPHOST1 OR APPHOST2
13. Ensure that the soapServerURL is set to internal virtual host <http://soainternal.mycompany.com> and the soapCallbackUrl to external virtual host <http://soa.mycompany.com> in both APPHOST1 and APPHOST2

If you are not using SSL for external access, both these properties should point to `http://soainternal.mycompany.com`

Ensure that the cluster flag is set to true

14. Restart both nodes.
15. Verify that the ESB DT container is down in one node and up in the other.

This completes the installation of AIA Foundation pack on a SOA cluster.

Steps for Deployment of AIA PIPS on AIA FP clustered deployment

Ensure that you have completed installing AIA FP on the SOA cluster before proceeding with this section

The following is the high level steps that we will be performing to complete the deployment of any given PIP on an AIA FP cluster installation (performed above) and is applicable for both Topology A and Topology B

We would execute a few install scripts to complete the installation on each node.

- On APPHOST2, we would execute the installation scripts manually. The scripts we execute would correspond to steps A and D of section 2.
- On WEBHOST2 and WEBHOST1 we would copy the required HTTP content into these hosts, covering step A of section 2 in each node.

This step is required only in the case of Topology A

- On APPHOST1, we would again copy all the AIA content and the install scripts to the disk. We would use the AIA Installer to cover all steps (A, B, C, D and E) mentioned in section 4.

Important: You need to follow the above actions in the same sequence

On APPHOST2

The following steps have to be followed on APPHOST2

1. Shutdown SOA Server on APPHOST1.
2. Ensure that APPHOST2 and WEBHOST1, 2 (in the case of Topology A only) is up and running.
3. In the deploy.properties file under AIA_HOME /config
 - Provide all information of the participating applications that your PIP uses. For example for O2C you will be providing Siebel and E-Business Suite information
 - Retain all other information, as it exists.
4. If you are installing a PIP XYZ, modify the following files in AIA_HOME /PIPS/Core/Setup/Install/XYZ
 - Execute the following command

```
grep "opmn:" *.xml"
```

- In the result if you see `http.hostname` use an editor to change it to `j2ee.hostname`. For example

```
"name="deployer.uri"
value="deployer:oc4j:opmn://${http.hostname}:${opmn.port}/${iasinstancename}/${oc4jinstancename}"
"name="deployer.uri"
value="deployer:oc4j:opmn://${j2ee.hostname}:${opmn.port}/${iasinstancename}/${oc4jinstancename}"
```

5. From `AIA_HOME/bin` directory execute the following command. (linux)

```
source aiaenv.sh
```

6. Execute the following scripts one after the other, while being in the directory `AIA_HOME/PIPS/Core/Setup/Install/XYZ`.

```
Ant --noconfig -buildfile Install.xml CopyAIAComponents.xml
```

7. This completes step A of section 2.
8. Now we will complete step D of section 2

```
Ant --noconfig -buildfile Install.xml ConfigDataSources.xml
ant --noconfig -buildfile Install.xml ConnectionFactory.xml
```

9. You should see connection Pools, data sources that are specific to your PIP.
10. This completes step D of section 4.
11. On `APPHOST1`, when you check the EM console, no configurations (connection pools, datasources etc.) should have been created.
12. In case you see any of them, manually undeploy them using the undeploy actions on EM.
13. Now bring down `APPHOST2`

On WEBHOSTn (Applicable only to Topology A)

We will copy the HTTP content created in the Apache folder on APPHOST2, to the real Apache folders on WEBHOST1 and WEBHOST2.

1. From the directory <SOA_HOME> /Apache/Apache/htdocs/AIAComponents/ on APPHOST2, copy the folders to the same directory structure under WEBHOST1.
 - ApplicationObjectLibrary> /<Application used by your PIP>
 - ExtensionServiceLibrary
 - UtilityArtifacts
 - Transformations, ApplicationBusinessServiceLibrary
2. Repeat the above 2 steps for WEBHOST2
3. Restart the HTTP server of both WEBHOST1 and WEBHOST2

On APPHOST1

The following steps have to be followed on APPHOST1

1. Shutdown SOA Server on APPHOST2
2. Ensure that APPHOST1 and WEBHOST1, 2 (in the case of Topology A only) is up and running.
3. In the deploy.properties file under AIA_HOME /config
 - Provide all information of the participating applications that your PIP uses. For example for O2C you will be providing Siebel and E-Business Suite information
 - Retain all other information as it exists.
4. If you are installing a PIP XYZ, modify the following files in AIA_HOME /PIPS/Core/Setup/Install/XYZ
 - Execute the following command


```
Grep "opmn:" *.xml"
```

 - In the result if you see http.hostname use an editor to change it to j2ee.hostname. For example

```
"name="deployer.uri"
value="deployer.oc4j:opmn://${http.hostname}:${opmn.port}/${iasinstancename}/${oc4jinstancename}"
"name="deployer.uri"
value="deployer.oc4j:opmn://${j2ee.hostname}:${opmn.port}/${iasinstancename}/${oc4jinstancename}"
```

5. Perform the following steps.

- Navigate to the folder AIA_HOME /util/DeployTool/BpelCopy
- Open the file build.xml and search for the string “Process deployment targets”. add “, bsrPublish” and save the file

```
<!------->
<!-- Process deployment targets -->
<!------->
<target name="process-deploy" depends="EditAIAConfigProperties,CustomiseWSDL,validateTask, compile,
deployProcess, deployTaskForm, deployDecisionServices, bsrPublish" />
```

- Navigate to the target bsrPublish replace http.hostname to j2ee.hostname
- In the AIA_HOME /util/DeployTool/ directory, open the file Tokenizer.properties

In this file, provide the values for the following three properties and save

- aia.home= <your AIA_HOME> /PIPS
- ESB_COPY_FOLDER = <your AIA_HOME> /util/DeployTool/EsbCopy
- BPEL_COPY_PATH= <your AIA_HOME> /util/DeployTool/BpelCopy
- Run the following command

```
ant -noconfig runbpel
```

6. From AIA HOME directory on APPHOST1, navigate to the folder PIPS/Core/Setup/Install/XYZ

7. From the command line (windows or unix based systems) execute the command

```
(linux) source AIA_HOME/bin/aiaenv.sh
ant -noconfig -f Install.xml > APPHOST1Install.log
```

8. Verify if all AIA processes were deployed as per AIA installation guide verification chapters. Verify if AIA Connection pools and AIA data sources are created.

9. Now bring up APPHOST2. Refer to the “Installing AIA Foundation Pack on Oracle SOA Suite Cluster Deployments” paper on how to start up SOA server. This should propagate all BPEL process to APPHOST2.

10. Verify that the AIA datasources and AIA connectionpool exist on APPHOST2.

This completes the deployment.

Complete the steps provided in the section below to perform Post Install Actions for PIP deployment.

Post Install actions for PIP deployments on AIA FP cluster

The following steps have to be followed on APPHOST1

1. If you were using RAC database, then you need to ensure that all AIA related datasources and connection pools point to the RAC URL.

For example for O2C2, this is EbizDB_Pool

- On the EM console, navigate to the JDBC resources of the oc4j_soa container and modify our PIP specific connection pool information as follows.
- Change the JDBC URL to RAC format. Typically this would be of the format

```
Jdbc:oracle:thin:@(DESCRIPTION=(LOAD_BALANCE=on) (ADDRESS=(PROTOCOL=TCP)(HOST=host1)
(PORT=1521)) (ADDRESS=(PROTOCOL=TCP)(HOST=host2)(PORT=1521))
(CONNECT_DATA=(SERVICE_NAME=service_name)))
```

- Repeat the same steps for both APPHOSTs

Note: If you are using XA and Distributed Transaction Processing (DTP) provide the corresponding DTP service name for each APPHOST in the in the Connection Pool on each APPHOST.

For more information on DTP and XA for SOA Suite, refer

<http://www.oracle.com/technology/tech/soa/pdf/oracle-soa-suite-xa-rac-guide.pdf>

2. Restart both nodes.
3. Verify that the esbdt container on one of the APPHOST is down and the rest all are up.

Upgrading AIA installations on SOA cluster

FP Installation should be upgraded first followed by PIP upgrade. While FP upgrade is a standard process PIP upgrade has to be done on a case-by-case basis depending on the content of the upgrade, customizations performed etc.

The steps for the PIP upgrade can be extrapolated from the explanation in this paper. Below are the steps to upgrade AIA FP Cluster installation.

Important: The following steps have to be performed in both APPHOST1 and APPHOST2

1. Launch the AIA installer as per instructions in the AIA Installation Guide. Provide the AIA Home directory of the existing AIA Installation.
2. Once you provide the AIA_HOME, the Installer will automatically detect the existence of a 2.3 FP Installation.
3. Click Install.
4. From the directory AIA_HOME/bin, execute the following command

```
source aiaenv.sh (linux)
aiaenv.bat (windows)
```

5. Navigate to AIA_HOME/patchset/Infrastructure/upgrade/scripts
6. Execute

```
ant --noconfig -f FP23to24Upgrade.xml copyFP23FilesToAIAHome >
$AIA_HOME/Infrastructure/install/logs/FP23to24Upgrade_files.log
ant --noconfig -f FP23to24Upgrade.xml DeployAIAFPCore >
$AIA_HOME/Infrastructure/install/logs/FP23to24Upgrade_deploy.log
```

7. This should upgrade your Foundation Pack installation on both APPHOSTS.

Conclusion

High Availability is a requirement in most cases where AIA is to be deployed on production servers. While this guide attempts to provide step-by-step instructions to deploy FP and PIPs to a SOA cluster, these steps can be modified/extrapolated to suit your requirement.

The steps in this paper provides repeatable and automated way of installing AIA processes/artifacts to a SOA cluster in addition to ensuring that all SOA and AIA configurations are setup correctly for a smooth functioning of AIA on a clustered environment.



Installing AIA on Oracle SOA Suite Cluster
Deployments

Sep 2009

Author: Arvind Srinivasamoorthy, Purushotham
Kola

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



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

Copyright © 2009, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.