

High Availability Patching for Fusion Middleware

*Oracle Maximum Availability Architecture White Paper
September 2011*

Maximum Availability Architecture

Oracle Best Practices For High Availability

Introduction	1
Artifacts in a Patch	2
Application JARs and EARs	2
Utilities	2
Composites	2
Patching Tools	3
OPatch	3
SmartUpdate	3
HA Topologies.....	3
Cluster Configuration	4
Shared Middleware Homes	4
Backup Middleware Homes	4
Replication Pairs	5
Rolling Patch Concepts	5
Patching Downtime: Shutdown vs. Restart	5
Graceful Shutdown	6
Shared MiddleWare Homes and Session Failover Configuration....	7
Shared Middleware homes and Central Inventory	8
Rolling Patching Example.....	9
Example Topology.....	10
Scenario 1: Using OPatch to patch Oracle WebCenter.....	11
Scenario 2: Using SmartUpdate to patch Weblogic Home	12

Introduction

In a High-Availability environment, the goal is to maintain a specified level of service at all times. The environment is one that is resilient to failures of individual components. The environment should also be able to provide a level of service during periods of scheduled maintenance. That is, the goal is to maintain this level of service while applying necessary patches.

This paper documents the best practices for applying one-off patches in a High Availability environment. The focus is on individual patches, for example those intended to address a specific bug. During the patching process, servers can be patched in a rolling manner; only a subset of the servers is down at any one time while other servers continue to serve requests.

This paper briefly covers the types of artifacts that will be applied in a typical patch and the patching tools – Opatch and SmartUpdate – used to apply one-off patches. Also covered is shared storage topologies where a Middleware home may be shared among machines, as well as the best practices for managing session migration and ensuring a graceful shutdown of managed servers. This paper does not address larger patchsets or upgrades.

Artifacts in a Patch

A one-off patch will contain a set of files as well as instructions on how to apply the patch. In this paper, we'll focus on the most common type of patch – a patch to Application JARs. But the different types of patches are all listed here.

Application JARs and EARs

A patch to fix a product feature will most often be in the form of a replacement to deployable application files. This will be a set of JAR or EAR files which will replace the files in Oracle Home. The files might be library files or an application EAR such as webcenter.ear.

In order for the patch to take effect these files need to be not only copied to the proper location but also deployed to the managed servers. This deployment can be rolling and can take the form of a server restart.

Utilities

Utilities are files which have no runtime dependency. This can be configuration utilities such as config.sh in the WebLogic directories or scripts such as configure-joc.py in the Webcenter Oracle Home.

Patching these files has no effect on runtime Availability and so they are not discussed further in this paper.

Composites

SOA Composites are applications which run inside the SOA infrastructure. They are not JEE applications. The SOA framework includes a versioning scheme so that different versions of the composite can run alongside each other.

Non-versioned soa composite patches are rare. In that case, existing tasks by users will become stale and will have to be re-initiated.

C Components

In the case of C artifacts which need to be patched, all processes which rely on the files should be shutdown while the patch is being applied and then restarted. This will include such applications as Oracle HTTP Server or Oracle Internet Directory.

Patching Tools

Oracle Fusion Middleware currently supports two tools for applying patches to a Fusion Middleware Home. OPatch is used to apply patches to a specific Oracle Home. Oracle SmartUpdate is used to apply patches to a WebLogic Home.

OPatch

OPatch is used to apply, maintain and rollback patches to an Oracle Home. The tool relies on the Oracle Inventory.

OPatch applies patches to only one Oracle Home at a time. For example, if an environment consists of a SOA Home and a WebCenter Home, OPatch will run with only one specific Home as the target.

SmartUpdate

SmartUpdate is the tool for applying patches to WebLogic. Patches are first downloaded to a staging directory. Then patches are applied by moving jars into a location where they will be picked up by the classpath. Servers need to be restarted to find the new Jar files.

HA Topologies

The first requirement for HA Patching is having a topology that can support it. An environment with only one Managed Server, for example, cannot be Highly Available.

There are further requirements that allow for greater availability when applying patches. They are covered in more detail in the next two sections on MiddleWare Homes and Clusters but are listed briefly here:

1. Clusters

Applications should be in a Managed Server cluster of 2 or more Servers. This allows one Server to be down while the other is serving requests.

2. Multiple MiddleWare Homes

In a shared storage environment, sharing a MiddleWare Home is one method of lowering management overhead. However, there should be multiple shared MiddleWare Homes to allow for redundancy.

3. Clusters span MiddleWare Homes

This is a further requirement of 1. and 2. above. All cluster members should not run from the same MiddleWare home. This is to allow greater redundancy.

4. Replication pairs span MiddleWare Homes

This is a further requirement on 3. above. Not only should clusters span MiddleWare Home but also replication pairs within a cluster should span MiddleWare homes. More detail on how to achieve this is below.

Cluster Configuration

Two or more managed servers are required in order to allow one managed server to provide service while the other server is down for maintenance.

This is only a minimum requirement. In practice, many servers should be available in order to provide availability as well as to be able to handle variations in the load. If one server is down among a cluster of N servers then the same load will have to be handled by only $(N-1)/N$ of the server resources.

Shared Middleware Homes

A shared Middleware Home is a Middleware home on shared storage that is used by more than one machine.

The advantage of sharing Middleware homes is that the number of Middleware homes that have to be maintained is significantly less than if one Middleware Home were assigned to each machine.

However, since the failure of a Middleware home, through corruption or through network issues, may affect servers running on multiple machines, there should be multiple MiddleWare Homes. The ideal number of shared homes will be a compromise between availability and maintenance.

Loss of a Middleware Home will affect multiple machines. If M machines are equally sharing N Middleware Homes, then loss of one Home may temporarily affect M/N machines meaning that only $(M-M/N)/M$ of the usual resources are available to handle the load.

Backup Middleware Homes

In addition to the active Middleware Homes, shared among one or more machines, extra copies or clones can be created that provide a backup for any Middleware Homes that become corrupted. These Homes can also be used as backups in case there are problems or

inconsistencies that arise during a patch apply. Creating these clones is a function of many storage servers.

Note that if a Middleware Home needs to be replaced with a backup, this will affect all servers running from that Home and incur the loss of service specified in the previous section.

Replication Pairs

In order to maintain session information in the event of server failure or downtime, servers can be configured to replicate the session information from one managed server (the primary) to another managed server (the secondary). If the primary server is unavailable, users will automatically be directed to the secondary server.

Selection of a secondary server is done automatically. The preference in selecting a secondary server is to favor servers that are running on other machines.

In the case of shared Middleware homes, secondary servers should be configured that also do not share a Middleware home. This is not handled automatically but can be configured using a feature known as Replication Groups.

In the next section, we'll walk through the steps to configure Replication groups that span Middleware Homes.

Rolling Patch Concepts

Before walking through an example of a Rolling Patch, there are a few other concepts that need to be understood ahead of time.

Patching Downtime: Shutdown vs. Restart

For most one-off patches, the patch can be applied while the servers are running. This is because most patches are patching application jars and these are not refreshed until the application is redeployed or the server is restarted.

This means that the only downtime that is incurred is after the patch has been applied while the server is quickly restarted. Multiple servers that share a Middleware home can also be restarted one at a time.

The table below shows what needs to be restarted or shutdown for different types of patches.

RESTART AND SHUTDOWN REQUIREMENTS

TYPE OF HOME	TYPE OF ARTIFACT	IMPACT ON SERVER LIFECYCLE
--------------	------------------	----------------------------

Oracle Home such as Oracle_SOA1, Oracle_WC1, etc.	Application jars or ears	All Servers of that type need to be restarted. If the patch is only to a WebCenter Home, for example, there is no need to restart SOA servers or those of other applications.
Oracle Home such as Oracle_SOA1, Oracle_WC1, etc.	Non-rollable application patch	See note below this table. In this case, all Managed Servers of that type need to be shutdown across the domain. This is not an HA usecase
Weblogic Home	Weblogic application jars	All Servers associated with this Middleware Home need to be restarted regardless of the application type.
Weblogic Home	Non-rollable weblogic patch	See note below this table. In this case, all Managed Servers of that type need to be shutdown across the domain. This is not an HA usecase

Note that in the table above, the type of artifact does not guarantee that a patch is rollable. There are certain application files which may not be rollable, often because they are a patch to a function of cluster communications and/or the patch is not backward compatible. That is, the type of patch does not support having an un-patched managed server run alongside a patched managed server.

In all cases, instructions in the patch README supersede instructions here or elsewhere in this paper.

Graceful Shutdown

When shutting down servers, a graceful shutdown should be used. This allows existing users some time to complete their work. In the case of sessions, the failover should occur regardless of how a server is shutdown since active sessions should already exist on a secondary server. But there may be other work, such as long-running transactions, which need to be completed.

A graceful shutdown first suspends the server from receiving new requests and then allows existing work to complete. To initiate a graceful shutdown from the Admin Control Panel:

1. Select the **Control** tab after selecting **Environment->Servers** from the left-hand pane.
2. Select the Server to be shutdown
3. In the **Shutdown** drop-down menu, select **When work completes**.

The disadvantage of a graceful shutdown is that some work may be taking an unreasonably long time to complete and may prolong server shutdown indefinitely. In order to prevent this, first set some parameters for the graceful shutdown:

1. From the left-hand pane, select **Environment->Servers**
2. Select the name of the Server to be configured
3. Select **Control->Start/Stop**.
4. Un-Select the box **Ignore Sessions during Shutdown**. This will ensure that existing sessions have been replicated on another server.
5. In the **Graceful Shutdown Timeout** box, enter the number of seconds that the server should wait before shutting down. If this box is left empty then the server may wait indefinitely. The exact value to enter here will depend on the system.

Shared MiddleWare Homes and Session Failover Configuration

For Fusion Middleware applications, in-memory session replication should be already configured for replication of session information within a cluster.

The default, as mentioned previously, is to replicate to servers that reside on another machine. The exact order in which a secondary server is chosen is illustrated in the table below:

CHOOSING A SECONDARY SERVER

SERVER RANK	SERVER ON ANOTHER MACHINE?	SERVER IN PREFERRED REPLICATION GROUP?
1	Yes	Yes
2	No	Yes
3	Yes	No
4	No	No

In this section, we'll configure a Replication group so that the preferred secondary server is one that does not share the same MiddleWare Home as the primary server.

Since a secondary server that does not share the same MiddleWare home will always be on another machine, the table becomes:

CHOOSING A SECONDARY SERVER

SERVER RANK	SERVER ON ANOTHER MACHINE?	SERVER SHARES MIDDLEWARE HOME?
1	Yes	No
2	Yes	Yes
3	No	Yes

To configure Replication Groups to favor replicating to servers that do not share a Middleware Home, follow these steps:

First, generate a unique name for each MiddlewareHome. For example, this can be MWHOME1, MWHOME2, MWHOME3...MWHOME(N).

Then, for each Server:, in the Admin Console

1. Select **Environment-Servers** and then the name of the Server to be configured.
2. Select the **Cluster** tab
3. In the field, **Replication Group**, enter the name of the MiddleWare Home which this server runs from, For example MWHOME2.
4. In the **Preferred Secondary Group**, enter the name of another Middleware Home, such as MWHOME3.
5. Save all changes.

For large numbers of Homes, MWHOME(N) can be configured to replicate to MWHOME(N+1) with the last MWHOME replicating to MWHOME1.

Shared Middleware homes and Central Inventory

Patching tools such as OPatch rely on the Oracle Inventory in order to know what patches have been applied. Since an Oracle Inventory corresponds directly with a MiddleWare Home, the oraInventory location should also be on the same shared storage as the MiddleWare Home.

In order to ensure that a shared Middleware home environment is correctly configured for patching:

1. Ensure that an inventory pointer exists on the local machine. This will usually be the file /etc/oraInst.loc

2. The contents of the oraInst.loc file will be a line pointing to the location of the actual Inventory. Here is an example:

```
Inventory_loc=/shared_vol/mwhome1/oraInventory
```

3. Ensure that the inventory location is on the same shared volume as the MiddleWare Home.

Patching Instance Homes

An exception to many of the above concepts is that of patching Instance Homes of products such as Oracle HTTP Server or Oracle Internet Directory. The lifecycle of these products is not managed by Oracle WebLogic but by Oracle Process Manager (OPMN) In addition, these products will often contain C components which should not be patched while the Instances are running.

To patch Instances of Oracle HTTP Server, for example, all HTTP processes should be shutdown and the patch apply rolled to all Instance Homes as follows:

1. Shutdown the affected instances using opmn.

```
$ opmnctl stopproc ias-component=HTTP_Server
```

2. Apply the Patch using OPatch

```
$ opatch apply
```

3. Restart the Instance

```
$ opmnctl startproc ias-component=HTTP_Server
```

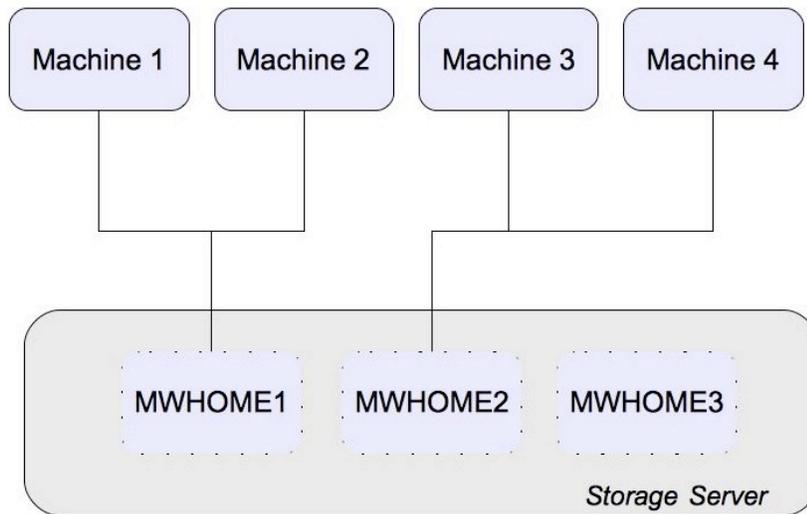
Repeat the above procedure for all affected instances. The 'stopproc' command is a graceful shutdown which will not interrupt requests in transit. To prevent new requests from arriving at an HTTP Server which is down, the front-end Load Balancer should be configured to remove this particular HTTP Server from the list of routable HTTP Servers **before** the HTTP server is brought down.

Rolling Patching Example

This section outlines the steps to apply a Rolling patch assuming that the requirements outlined in the previous sections have been met.

Example Topology

For the purposes of this example, we'll assume 4 machines sharing a total of 2 Middleware homes. Additional Middleware Homes may exist for recovery purposes as well.



Each machine has one Middleware Home mounted as in the diagram above. On each machine we have at least two Managed Servers running – a WebCenter Spaces server and a SOA server.

Machine 1

Running: Oracle_SOA1, Oracle_Spaces1, AdminServer

Machine 2

Running: Oracle_SOA2, Oracle_Spaces2

Machine 3

Running: Oracle_SOA3, Oracle_Spaces3

Machine 4

Running: Oracle_SOA4 , Oracle_Spaces4

Using this environment, we'll run through a rolling apply for the scenarios below.

Scenario 1: Using OPatch to patch Oracle WebCenter

Using the above environment, this section walks through an opatch-apply of a patch which:

- 1) Affects WebCenter Spaces only
- 2) Does not affect the SOA Servers
- 3) Only Requires Server Restart. That is, the patch itself can be applied while the servers are running.

Patch first Middleware Home

In this case, we are only patching a WebCenter Home. When running OPatch the Oracle Home can be specified on the command line or it will default to the Oracle Home from where OPatch is being run.

Before applying a patch, a backup of the current Home should be available. This can be a previous backup or storage server clone – such as MWHOME3 in the diagram above.

The steps in applying a patch involve first downloading the patch from Oracle Support and then, after ensuring that OPatch is in your system path, running OPatch from the patch directory.

OPatch comes with a `-report` option which will run through an apply without actually performing any actions. It can be run as follows:

```
$ ./opatch apply -report
```

When everything looks satisfactory, apply the patch:

```
$ ./opatch apply
```

This will create a log file and print messages to standard output about the files which have been changed. In our example patch, here are the relevant messages about the files being moved:

```
Patching component oracle.webcenter.application, 11.1.1.4.0...
Copying                               file                               to
"/u02/product/fmw/Oracle_WC1/archives/applications/webcenter.ear"
Copying                               file                               to
"/u02/product/fmw/Oracle_WC1/lib/java/internal/oracle.webcenter.spaces/11.1.1.0.0/webcenter-app-core-model.jar"
```

```
Copying file to
"/u02/product/fmw/Oracle_WC1/webcenter/modules/oracle.webcenter.sp
aces_11.1.1/oracle.webcenter.spaces.fwk.ear"

ApplySession adding interim patch '10349822' to inventory
```

Rolling Restart of Middleware Home1 Spaces servers

Only the Webcenter Spaces servers need to be restarted. Even though there are SOA servers running from this MiddleWare home, the patch is a Webcenter Patch and only affects the Webcenter Home.

Restart the Webcenter Spaces servers as follows:

1. Perform a graceful shutdown of Spaces1 on Machine1
2. Start up Spaces1 on Machine1
3. Perform a graceful shutdown of Spaces2 on Machine2
4. Start up Spaces2 on Machine2

Patch second MiddleWare Home

Follow the same steps as above in performing the same patch onto the second MiddleWare Home.

Rolling Restart of Middleware Home2 Spaces servers

Restart the Webcenter Spaces servers as follows:

1. Perform a graceful shutdown of Spaces3 on Machine3
2. Start up Spaces3 on Machine3
3. Perform a graceful shutdown of Spaces4 on Machine4
4. Start up Spaces4 on Machine4

Scenario 2: Using SmartUpdate to patch Weblogic Home

Using the above environment, this section walks through an apply of a patch which:

- 1) Affects WebLogic and thus also Webcenter and SOA
- 2) Only Requires Server Restart. That is, the patch itself can be applied while the servers are running.

Patch first Middleware Home

The Oracle Weblogic Server is a foundation for Fusion Middleware applications. Any patch to Oracle Weblogic will thus affect servers running from all Oracle Homes.

Currently, Weblogic Server patches are downloaded from Oracle Support and then applied using SmartUpdate.

Patches are first downloaded to a patch directory. Then, the bsu utility can be used to examine all available patches as follows:

```
$ ./bsu.sh -view -status=downloaded -  
prod_dir=/u02/product/fmw/wlserver_10.3
```

where prod_dir is the Weblogic home directory.

Patches can then be installed using a command such as the following:

```
$ ./bsu.sh -prod_dir=/u02/product/fmw/wlserver_10.3 -  
patchlist=GGXK -verbose -install
```

Rolling Restart of Middleware Home1 servers

In this case, all servers, including the AdminServer, that run from Middleware Home 1, need to be restarted:

1. Perform a graceful shutdown of AdminServer on Machine 1
2. Startup AdminServer on Machine 1
3. Perform a graceful shutdown of Spaces1 on Machine 1
4. Startup Spaces1 on Machine 1
5. Perform a graceful shutdown of SOA1 on Machine 1
6. Startup SOA1 on Machine 1
7. Perform a graceful shutdown of Spaces2 on Machine 2
8. Startup Spaces2 on Machine 2
9. Perform a graceful shutdown of SOA2 on Machine 2
10. Startup SOA2 on Machine 2

Patch second MiddleWare Home

Follow the same steps as above in performing the same patch onto the second MiddleWare Home.

Rolling Restart of Middleware Home2 servers

In this case, all servers that run from Middleware Home 2, need to be restarted:

1. Perform a graceful shutdown of Spaces3 on Machine 3
2. Startup Spaces3 on Machine 3
3. Perform a graceful shutdown of SOA3 on Machine 3
4. Startup SOA3 on Machine 3
5. Perform a graceful shutdown of Spaces4 on Machine 4
6. Startup Spaces4 on Machine 4
7. Perform a graceful shutdown of SOA4 on Machine 4
8. Startup SOA4 on Machine 4



Oracle White Paper Title: High Availability
Patching for Fusion Middleware
September 2011
Author: Richard Delval
Contributing Authors: Pradeep Bhat, Michael
Blevins

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 © 2011, 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.