Oracle Maximum
Availability Architecture

Oracle Commerce

Software Planned Maintenance

ORACLE WHITE PAPER | APRIL 2016

Table of Contents

Introduction

Patching and maintaining software are critical application management tasks for maintaining a healthy and productive Oracle Commerce solution. The process of patching itself can present different challenges that can potentially increase the work and time involved in each patching exercise. Issues could arise, such as patch conflicts, not meeting required prerequisites, and even unnecessary downtime. Spending the time to set up a patching strategy can save time and effort and prevent errors when patching a production Oracle Commerce environment. This white paper provides an overview of the recommended patching strategy for Oracle Commerce. This information is intended as a guideline for maintaining a patched and highly available Oracle Commerce environment and may need customization to accommodate requirements of an individual organization.

Oracle Commerce Components

The following is a list of the components that may require patching for an Oracle Commerce system. It includes both software and hardware operating systems.

Oracle Java Development Kit (JDK)

Oracle Commerce is a Java-based suite of applications. While the Guided Search applications are installed with an embedded JDK, the Commerce Platform applications are not. The Commerce Platform applications require a version of the JDK outlined in the Commerce Supported Environment Matrix. See MOS article 1345041.1 for more information.

Oracle Commerce Suite

Commerce Platform is the base installation from which applications are built. The base installation directory is also known as `DYNAMO_ROOT`, from the ATG Dynamo application, which is the retired legacy application. The base application contains the `DYNAMO_HOME`, which contain the scripts to build and maintain the applications, and contains the server configuration layers, which are built into the application packages. Oracle Commerce Suite includes B2C and B2B applications, Business Control Center (BCC, which contains Content Administration and Merchandising), Commerce Service Center (CSC), and Commerce Reference Store (CRS). Support software may also include Oracle Business Intelligence (BI), Outreach, and Knowledge.

Oracle Commerce Guided Search and Experience Manager

Guided Search is the search functionality of the Oracle Commerce Suite. It is joined by the Experience Manager application, which is used to set up how pages are displayed with the search results. It can also be used for general web page look and feel. Guided Search and Experience Manager are in separate installation locations from the Commerce Platform applications.

Oracle Fusion Middleware (FMW) Applications

Oracle Commerce allows for the deployment of its applications onto several different application server types. Among these are Oracle WebLogic Server, for running the applications, and Oracle Coherence, for external repository and HTTP session caching. Both WebLogic Server and Coherence are members of the Fusion Middleware family of products. See MOS article 1345041.1 for more information about supported application server versions for different Commerce application versions. The Oracle GoldenGate Management agent installed in the MAA environment for monitoring and controlling GoldenGate processes also uses the Fusion Middleware base installer for the standalone Java Agent.

Oracle Clusterware and Oracle Database

Oracle Clusterware is the control center of all database nodes within a Commerce implementation. Oracle Database contains the Commerce repositories and all other data associated with the Commerce applications. Oracle Database is normally deployed on several servers, known as nodes, across either LAN or WAN configurations for high availability.

Oracle Engineered Systems

Oracle Commerce applications, Oracle Clusterware, and Oracle Database can be deployed on Oracle Engineered Systems or commodity hardware. Oracle Clusterware and Oracle Database software are typically deployed on Oracle Exadata Database Machine, and Commerce, WebLogic Server, Coherence, and Java Development Kit software are typically deployed on Exalogic Elastic Cloud Engineered Systems. This hardware is included in this document because scheduling Exadata image updates and Exalogic operating system upgrades also require planning.

Define Business Requirements

Before a proper patching strategy can be defined, it is important to have specific details defined that may vary across companies. These details are part of defining the high availability requirements. Defining your business requirements is an important task that helps determine the patching strategy and should be done prior to the definition and documentation of a patching strategy. Defining the high availability requirements of the Commerce system involves the following:

1. Performing a business impact analysis which consists of categorizing the business processes based on the severity of the impact from an IT-related outage.
2. Identifying and categorizing the critical business processes in Commerce that have high availability requirements (i.e. monitoring of mission critical systems).
3. Calculating the length of downtime (both planned and unplanned) as it pertains to the time of day, day of week, frequency, and business constraints.
4. Defining the recovery time objective (RTO) and the recovery point objective (RPO). RTO is the maximum amount of time that Commerce could be down before the company suffers material losses. RPO is the maximum amount of data Commerce could lose before the company suffers material loses (i.e. data loss).
5. Understanding the total cost of ownership (TCO) and the return on investment (ROI).

The *Oracle Database High Availability Overview* provides a good explanation of this process. Once the above items are defined, it is easier to understand the required level of availability for the Commerce environment. For details about High Availability for Commerce, refer to the Oracle MAA Best Practices site for Oracle applications.

Patching Strategy Overview

Patches are critical in order to maintain a healthy Commerce environment. Developing a proactive patch strategy helps avoid issues and allows administrators and users to take advantage of new technology and features. Patching is not a one-step process but involves a strategy around timing, testing, the actual patching steps, and any possible recovery or fallback options. The list below details some of the tasks involved in developing proper patching strategies.

- Determine the patching strategy that covers the requirements of your organization.
- Determine the appropriate timing.
- Prepare for the change.
- Incorporate sufficient testing.

Types of Patches

Commerce Release

A Commerce Release is a major or minor release to the Commerce Suite and Guided Search and Experience Manager set of applications. Each release may require database DDL changes. Check the release notes to verify whether there are any required DDL changes. Rebuilding all of the application EAR packages for deployment may also be required. Each Commerce Release patch is governed by the supported environment's matrix found in MOS article 1345041.1. This matrix shows the supported versions of application servers, JDK, databases, and other support software for each given Commerce Release version. For more information about the Commerce patching process, please refer to MOS article 1517903.1 and MOS article 1486966.1.

Patch Sets

A patch set is a full release of the product and is applied using the upgrade process. It is a point release, such as upgrading the Middleware applications from 12.1.2 to 12.1.3. It includes patches for all middleware components including WLS and Coherence. A patch set release of Fusion Middleware applications can be released as often as every 9 months.

Patch Set Updates / Security Patch Updates (Critical Patch Updates)

Patch Set Updates (PSU) are cumulative patches containing critical recommended bug fixes and security fixes, and usually contain updates to all Fusion Middleware components including WLS and Coherence. These patches are intended to be low-risk, so they do not include fixes that require re-certification or configuration changes. For more details, please refer My Oracle Support note Patch Set Updates for Oracle Products (Doc ID 854428.1).

The Security Patch Updates (formerly known as Critical Patch Updates) are cumulative patches that limit the included bug fixes to security fixes only. Starting with Oracle Database version 12.1.0.1, Security Patch Updates will no longer be available. The PSU is Oracle's preferred proactive patching mechanism. Both of these patch types are released on a quarterly schedule: the Tuesday closest to the 17th of January, April, July, and October.

One-off Patches

One-off patches are bug fixes provided as required for a single bug fix or a collection of bug fixes. They may also be created for customer-specific security bug fixes. A one-off patch could also be a diagnostic patch that is created to help diagnose or verify a fix or collection of bug fixes. The recommendation is to apply the bundle patch (described below) and not apply one-off patches unless recommended by Oracle Support. Before applying a one-off patch:

- Make sure that the patch is approved by Oracle Support to work with the current Commerce release and that it is required to address a current issue/bug.
- Refer to the patch README to carefully set the `ORACLE_HOME` environment variable to the component that you patch. In some cases, it may be the middleware home (`MW_HOME`) depending on the patch in question.
- One-off patches for the Commerce Platform installation are typically Java archive (`JAR`) files, which are placed in a location inside of `DYNAMO_HOME` and assembled with the Commerce Platform's `runAssembler` binary, with the `-prepend-jars` option. Refer to a patch's README file for information about deployment.
- One-off patches for the Commerce Platform installation can also be static text files, which need to be placed inside the server configuration layers before a new application Java archive (`EAR`) file can be built. Again, refer to the patch's README file for information about deployment.

Bundle Patches

In order to reduce the number of patches and to assist in the overall availability of the Commerce system, Oracle is now releasing patch bundles for the following software:

- Oracle Fusion Middleware
- Oracle Clusterware and Oracle Database

These bundles are released on a monthly basis and contain the latest PSU and any critical one-off patch for that release (up to the cut off window for the patch creation). Each bundle patch is cumulative for that particular release and is marked as a "Recommended Patch" on a quarterly basis.

At this time, the bundle patch concept does not apply for the following components and one-off/PSU patches will still be created:

- Commerce Platform specific patches

Rolling / Non-Rolling Patches

Rolling/Non-Rolling Patches does not refer to specific patch types, but the method by which patches are distributed to different hosts, nodes, or instances.

- Rolling: The environment can be left running, and a single host, node, or instance is updated at a time automatically. Most of the Oracle Clusterware and Oracle Database patches now have a built-in mechanism, so that the patch is initiated on one member, and it will read cluster information to discover all members of the cluster and patch each member individually and sequentially.
- Non-Rolling: These patches must be manually deployed onto each host, node, or instance. JDK, Commerce Suite, FMW, Exadata image, and Exalogic OS patches are all non-rolling.

Determine Patch List and Required Patching Steps

The available patching options may vary for the different Commerce components, and certain patches may require downtime for the entire Commerce environment. The table below indicates downtime for the entire Commerce environment based on the component to be patched or upgraded.

TABLE 1: COMMERCE DOWNTIME BASED ON COMPONENT

Component	Environment Outage	Comments
JDK	MAYBE	The structure of some serialized objects may change, which may cause errors when instances running different patch levels try to communicate. Refer to JDK patch README for details to ensure that the objects have not changed.
Commerce Suite	MAYBE	DDL changes may force an entire database outage, taking the environment offline, even if a standby database is used for HA. Application package deployment may not be able to be deployed in a rolling manner. Refer to the patch README to ensure that changes do not need to be made which may force a database outage.
Commerce Guided Search / Experience Manager	YES	Baseline indexing of objects to be searched must be performed. This is a complete re-indexing which will temporarily suspend searching capability.
Fusion Middleware applications	MAYBE	This is dependent on the environment setup. In a fully active/active configuration, an entire environment can remain running while the other environment is patched. For smaller configurations, if all instances are on the same host, all instances on that host need to be shut down to run the patch. In order to ensure no downtime of applications, redundant applications must be present and patching must be performed in a rolling manner.
Clusterware and Database	NO	Most patches can be performed in a rolling manner.
Exadata images and Exalogic OS	NO	Most patches can be performed in a rolling manner.

Best Practices:

- The list of Commerce product availability and patch upgrade paths can be found in MOS article 1356457.1 and the Commerce patching policy can be found in MOS article 1517903.1
- Set a patch window appropriate to the defined high availability requirements of the Commerce environment.
- Determine the patching procedures/process in advance that will meet the required RTO/RPO for Commerce.
- Organize the patches according to the component they apply to and the downtime requirements for each patch.

It is important to remember that the patching plan should include back-out (rollback?) steps for any patch and a good backup and recovery plan in case something goes wrong during a patching window. Each patch includes the back-out steps for that patch.

Planning

What timing works for your organization? It is important to note that some patching may occur out of cycle. For example, there may be a required patch for the database (i.e. security patch) that is not specifically required by Commerce. In such cases, patching of the Commerce databases may fall under this patch cycle but not be considered part of the standard Commerce patching window.

- *Monthly?* This is probably too aggressive for any organization. A monthly patching schedule could be resource intensive as it would require a resource to spend a majority of their time during the month on preparing, testing, and patching the test production systems. Once this is done, it would be time to start over again on the next set of patches.
- *Quarterly?* A good strategy; however, it may not be feasible in some organizations due to resource availability. This timing may be appropriate for companies with smaller Commerce environments (100 FMW hosts or less) or specific company policies requiring more frequent patching.
- *Semi-annually?* Probably the interval that provides the best balance between the TCO and keeps systems current on patches. This is the recommended patch window. With a semi-annual patch plan, both the test and production environments are patched twice a year. This actually means patches are applied on a quarterly basis as patches are applied to the test environment first. (See Table 2: Sample Semi-Annual Patch Window below).
- *Annually?* For most environments, this is most likely too long and they will find a need to stay more current on bug fixes and newer plug-in updates.

Best Practices:

- Set up a patch window based on patching the production Commerce environment semi-annually. This means patching the test system(s) first, and allows approximately 2 months of testing before application to production. Of course, this window can be shortened if 2 months is not required by moving the production patching window.
- Sign up for patch and security alerts from Oracle so you get notifications of patch releases and important security fixes.

Table 2: Sample Semi-Annual Patch Window

Env	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Test	X						X					
Production				X						X		

Prepare for the Change

Research the patches. Understand what is changing in the patch application steps and any required pre- or post-application steps. Oracle's patching options change as more and more patches can be applied online or in a rolling manner. Be sure to leverage these patch application improvements.

Understand the patches that will be applied, the amount of risk or change that will be introduced into the environment, and therefore, the amount of testing that should be performed on the test environment. There is a tradeoff between high availability and the overall time it takes to apply patches to a Commerce environment. For example, depending on the amount of downtime acceptable for your Commerce environment it may take less time to take the downtime and apply the patches to all WebLogic servers at the same time. If high availability is a concern, then always choose the online or rolling patch method, where possible, which extends the time to complete the patching.

Best Practice:

- Understand the amount of change that the patch will introduce into the environment to determine the amount of testing required. This is determined by factors such as components that are being patched, the number of fixes in the patch, and the type of patch. For example, understand if the patch contains only bug fixes or includes modified or new functionality (i.e. a Commerce Platform Release or FMW, Clusterware, Oracle Database patch set will include more change than a simple one-off bug fix).

Testing

Testing the list of required patches on a test Commerce environment provides better benchmarks for the amount of downtime to expect (if any), the amount of time for the patching window (which may or may not change), the list of patches to apply, and the exact steps involved in order to meet the required RTO. The test environment should be configured as close to the production system as possible so that the testing will more closely replicate the production environment. The most challenging issue is creating the "load" on the Commerce environment from a workload testing software suite, such as Oracle Application Testing Suite (OATS). One option for accomplishing this includes creating a separate environment for all test and/or development systems. This allows for a staged patch implementation cycle, and patches are well tested before affecting production. For companies that prefer to have all of their systems contained in a single Commerce environment, the Commerce test system may not have as many instances, and therefore will not have an equivalent activity/load to the production Commerce applications.

Testing patches before going live is critical, which is why a test system for the Commerce environment is so important, especially if the environment requires a higher level of availability to meet the required levels for RTO and RPO. Testing should not only check to see how the patch behaves during installation, but also how it behaves when rolled back. It is also important to define a test plan that tests for all of the critical processes – technical and functional. It is recommended that you define a list of test cases appropriate for that application or organization. Table 3 provides some guidelines for the testing level for the different patch types. The application and rollback of the patches should be tested for all patch types. This may involve different levels of testing as described below.

- Bug verification: If a patch is applied for a specific bug fix, then verify that it does fix that bug.
- Basic testing: This testing includes verifying the overall functionality of all Commerce, FMW, Clusterware, Oracle Database, and OS applications. Basic functionality should also be tested, including making sure all components stop and start successfully, and verifying the health of the Commerce components after the patch application.
- Full testing: More testing should be involved in the patch set releases, because these could include major changes for the Commerce applications. It is important to check as much functionality and performance as possible. This level of testing includes making sure all components stop and start successfully, verifying the health of the Commerce components, verifying all functionality of all components and applications, and verifying the application performance.

Table 3: Commerce Patch Testing shows the type of testing that should be performed for different patch types.

Table 3: Commerce Patch Testing

	Commerce Releases	Patch Sets	PSU/SPU	Bundle Patches	One-off Patches
Install / Rollback	Yes				
Bug Verification	Where applicable and possible				
Commerce Functionality	Full	Full	Basic	Basic	Specific for a bug
OS Functionality	Full	Full	Basic	Basic	If applicable

Best Practices:

- If multiple components are to be patched, patching one component at a time should be performed. Once the first component is patched and tested, then the next component can be patched and tested.
- Maintain a copy of Commerce for test purposes (maybe one that hosts dev/test systems for the environment).
- Ensure that the test system has the same patches as production prior to testing the new patches.
- Back up the environment before applying the patches.
- Pre-patch check: Run diagnostics, such as `exachk`, to ensure that all patch prerequisites are met.
- Download the patches and apply them to the test environment, checking for any patch conflicts. If conflicts exist, request merge or overlay patches.
- Post-patch check: As with the pre-patch checks, run diagnostics, such as `exachk`, to ensure that all patching is complete and error-free.
- Use the patch testing and patch apply details to determine the appropriate method for the patch application (high availability vs. patch window duration).
- Apply the patches to production, repeating the pre- and post-patch checks.
- For non-rolling patches to several applications simultaneously, use multiple windows to minimize downtime. This does not apply to applications, which are set up to be active/active. In this case, you would patch one application at a time to eliminate downtime.
- Ensure that all application packages to be deployed are staged prior to beginning patching. This applies to configurations where a central file share is used, where packages are stored on individual hosts, or where they are deployed to each managed server.

Example Patching Steps

This process does not cover the application of patch sets because those are applied using the upgrade process, which is well defined in Oracle's Upgrade Guide. The patching steps are broken down by component because a patch window may not always contain patches for all Commerce components.

1. Create a list of patches required. Please refer to MOS article 1356457.1 for information about Commerce Suite patches and MOS article 854428.1 for information about all other Oracle software components pertinent to the patch. Note that if a special one-off patch was applied, it may conflict with a patch on this list, but any conflict should have been discovered during the testing phase, and therefore an overlay or merge patch added to the list to resolve the conflict.
2. Document each patch to apply and the corresponding patch application steps.
3. Create a recovery plan, and be sure to document the patch back-out or rollback steps for each patch. Refer to the README file to determine steps for backing out each patch.
4. Organize and group the patches by component (i.e. JDK, Commerce components, FMW applications, etc). Download all patches and stage patches on servers.
5. Within the categories above, group together according to the patch apply requirements (rolling vs. downtime).
6. Before patching begins, stop the standby for the Commerce database, if applicable.
7. Back up the environment by taking a backup of the Oracle inventory location, `DYNAMO_HOME`, Middleware home, and WebLogic domain home, and either take a full backup or create a restore point for the Commerce database. For information about restore point and flashback database, see the Using Flashback Database and Restore Points section of the *Database Backup and Recovery User's Guide*.
8. Follow the steps in the README to patch the Commerce installation (if this is a Commerce Release patch).
9. Be sure to follow the patch application steps, including any prerequisite checks. For FMW, Clusterware, and Oracle Database patches this will be the `-analyze` option for the `opatchauto` command.
10. Follow the steps below when applying FMW, Clusterware, and Oracle Database patches. These steps are organized to provide the most uptime for Commerce.
    a. Execute all patch analyze steps prior to shutting down the WebLogic and Coherence servers.
    b. Execute the patch apply for the Bundle patches for the WebLogic and Coherence server (PSU patch and plug-in bundle patch).
    c. Shut down all WebLogic and Coherence servers (if this is a Commerce Release patch, only patching FMW applications can be done in a manually rolling manner).
    d. Apply all patches to the WebLogic and Coherence servers.
    e. Create new Commerce application EAR packages (if this is a Commerce Release patch).
    f. Deploy the new Commerce application EAR packages to the appropriate WebLogic servers (if this is a Commerce Release patch).
    g. Run scripts provided in the patch to update the Commerce database DDLs.
    h. Restart all WebLogic and Coherence servers.
11. Follow the steps below when applying Clusterware and Oracle Database patches.
    a. Before applying patches to the Commerce database, if flashback database has been enabled it is recommended that you take a restore point to be used in the event that a recovery is required. Please refer to the Using Flashback Database and Restore Points section of the *Database Backup and Recovery User's Guide* for details about using flashback database and setting restore points.
    b. If there is a standby Commerce database, in addition to the primary Commerce database, the standby should be patched first. See MOS article 1265700.1 for details.

c. Apply the latest PSU to the standalone or Oracle RAC primary Commerce database as per the patch README instructions. Note that these patches can be applied in a rolling manner so the database does not have to be shut down.

d. Apply any required one-off database patches.

e. Execute all required post-patch scripts for the repository patches.

f. Restart the recovery (real time apply) process for the standby repository database.

12. Follow the steps below when applying Exadata image patches (if applicable).

    a. Run `exachk` to see if all prerequisites are met.

    b. Follow the README of the image patch to run the `patchmgr` utility to patch the storage cells and InfiniBand switches. This can be run in a rolling or non-rolling manner. It is recommended that you run the update in a rolling manner.

    c. Follow the README of the image patch to run the `dbnodeupdate.sh` script to patch the database node servers. This can be run in a rolling or non-rolling manner. It is recommended that you run the update in a rolling manner.

If desired/required, test the disaster recovery site by performing a switch-over to the standby site and confirming the functionality. Depending on the company's strategy/requirements, the systems can remain running on the standby site or a switchback can be performed.

For any issues encountered during the patch application or found after patching the environment, a partial or full recovery of the environment may be required. Follow the defined back-out plan. If specific patches fail then they can be rolled back individually.

- Restore the backup of the Commerce `DYNAMO_HOME` server.
- Restore the backup of the FMW servers.
- Restore the database to the point in time prior to the start of the patching using a flashback restore point or the database backup.

Conclusion

Patching a Commerce environment is required to keep the tool current on fixes, enhancements, and functionality. As there are several components which make up a Commerce environment or set of environments, it is crucial that adequate planning be performed at all stages of an upgrade, whether it be a simple JDK upgrade or a full upgrade including Commerce applications, Fusion Middleware applications, Clusterware, and Oracle Database applications. Developing the right patching strategy will ensure a smoother patching experience while adding to the higher availability and dependability of the environment.

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

ORACLE

CONNECT WITH US

blogs.oracle.com/oracle
facebook.com/oracle
twitter.com/oracle
oracle.com

Integrated Cloud Applications & Platform Services

Copyright © 2016, 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 and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group. 0615

Oracle Commerce Software Planned Maintenance
April 2016
Author: Chuck Boucher
Contributing Authors: Bethany Lapaglia

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