

Critical Patch Update Implementation Best Practices

An Oracle White Paper
December 2006

Critical Patch Update Implementation Best Practices

Introduction	3
Executive Summary	3
Overview	3
Goals	4
Reference	5
About CPUS	5
CPU Contents	6
CPU Implementation Phases.....	6
Planning.....	8
Test Deployment	9
Production Deployment	10
Optimizing CPU Implementation	12
Environment Checker Tool	12
Testing Strategy	12
Cookbook	13
Merge Application Patches.....	13
Shared Product Homes	13
Choose The Right Tools.....	13
Proactive Maintenance	14
Conclusion.....	14

This white paper is intended for IT management and senior staff responsible for the planning and implementation of changes to Oracle systems or environments in their enterprises.

INTRODUCTION

This white paper is intended for IT management and senior staff responsible for the planning and implementation of changes to Oracle systems or environments in their enterprises.

This white paper describes best practices when implementing an Oracle Critical Patch Update (CPU) for enterprise customers running Oracle. After the executive summary, the first section of the document provides general information about CPUs. The next section provides a flow diagram of CPU implementation followed by details on planning, testing and deploying CPUs. The last sections of the document discuss possible optimization of the process and its benefits.

EXECUTIVE SUMMARY

Overview

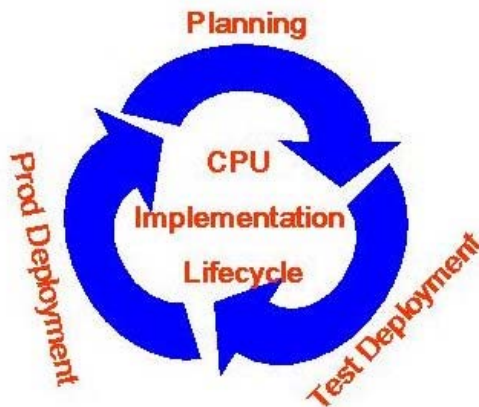
An Oracle Critical Patch Update (CPU) is a bundle of patches released on a quarterly basis to provide security fixes for Oracle products. Each CPU implementation consists of three important phases:

- Assessment & planning
- Test deployment
- Production deployment

The following diagram shows how these phases are interrelated:

For more information about CPUs, see the following documentation located on *OracleMetaink*:

- Oracle CPU FAQ
- Oracle CPU Program General FAQ
- FAQ for Oracle Security Alerts and CPU



This white paper describes these phases, providing a detailed flow diagram and tips for each of the phases. The other important aspect of this white paper is to provide

optimization guidelines that individual IT organizations can use for efficient CPU implementation. Guidelines for optimizing the CPU implementation are:

- Share product home across multiple middle-tier installations to apply each CPU only once.
- Merge multiple Oracle E-Business Suite patches into single bundle.
- Develop the right testing strategy to ensure system integrity when a CPU is implemented.
- Create a cookbook for CPU implementation customized for your IT environment.
- Proactively maintain environments to optimize CPU support.
- Develop the right tools and processes for health checks before and after implementation.

These points are explained in more detail in the Optimizing the CPU Implementation section.

Goals

Regular CPU implementation provides a number of benefits to customers, including:

- Allowing enough time for planning, rather than having unannounced and unplanned patch notifications, such as a security alert.
- Generally lowering cost as each CPU consolidates multiple security fixes into one patch. Customers do not have to apply many one-off patches at different times nor do they have the cost of bringing down systems and repeat testing.

CPUs continue to be improved to minimize the cost (in both downtime and labor) and the risk of implementing a CPU.

Oracle On Demand implements CPUs for hundreds of its hosted customers in a matter of days after the CPU is released. Many of the recommendations discussed in this white paper are the outcome of Oracle's experience of implementing CPUs in the On Demand business. This white paper describes several best practices and further opportunities to optimize CPU implementation. The resulting benefits include:

- Enabling customers to rapidly implement CPUs at a reduced cost and low risk.
- Enhancing the experience of those involved in CPU planning and implementation.

Oracle On Demand implements CPUs for hundreds of its hosted customers in matter of days after the CPU is released. Many of the recommendations discussed in this white paper are the outcome of Oracle's experience of implementing CPUs in the On Demand business.

- Lessening the impact of a CPU implementation on business and end users by reducing the total downtime required for CPU implementation.

Reference

Oracle On Demand is a good example of the rapid uptake and successful implementation of CPUs. Oracle On Demand has realized substantial cost savings and benefits by adapting many of the best practices and recommendations listed in this document. At the same time, Oracle On Demand customers have witnessed a reduction in downtime and greater consistency and reliability of CPU implementation.

From 2005 on, CPUs are the primary means of releasing security fixes for Oracle products.

ABOUT CPUs

A CPU is a bundle of patches released on a quarterly basis to provide security fixes for Oracle products. Since 2005, CPUs are the primary means of releasing security fixes for Oracle products. CPUs are released every quarter on a pre-announced date. CPUs are issued for vulnerabilities in supported products that compromise:

- Data confidentiality, i.e. an attacker is able to view information that he or she should be prevented from viewing.
- Data integrity, i.e. an attacker is able to modify information that he or she should be prevented from modifying.
- System availability, i.e. an attacker is able to disrupt legitimate use of or access to a system.

CPUs fix vulnerabilities that compromise the confidentiality, integrity or availability of a system.

Oracle releases one consolidated patch per version of the Oracle Database, Application Server, and Enterprise Manager. Collaboration Suite is similar, but fixes are provided per component. Where possible, Oracle consolidates E-Business Suite fixes into one patch. However, each CPU typically has more than one patch for each Oracle E-Business Suite release. For example, Oracle issues one CPU patch for each Oracle Database Release 9.2.0.6 and Oracle Application Server Release 9.0.4.1 (middle tier), and several one-off patches for Oracle E-Business Suite Release 11.5.10. PeopleSoft and JD Edwards patches are also currently provided as one-off patches.

CPU patches are cumulative for database and middleware products. This means that the database or middleware patch in the latest CPU include fixes for all earlier CPUs unless stated otherwise. However, CPU patches for Oracle E-Business Suite are not cumulative. PeopleSoft and JD Edwards patches are currently not cumulative, but will be in future. The CPU documentation for each Oracle product suite identifies whether the associated patches are cumulative or incremental in nature.

CPU Contents

Each CPU contains security patches and supporting documentation, including a documentation roadmap. The documentation roadmap helps you select the right patches for your systems and the correct supporting documents. There are three levels of documents included in CPU:

- **Level 1:** *Oracle CPU Advisory* contains information related to security fixes, including the risk assessment matrices. These matrices provide a list of components affected by the new security fixes. In October 2006, Oracle switched from a proprietary method for indicating the relative severity of security vulnerabilities in the risk matrices to the Common Vulnerability Scoring System (CVSS). The advisory lists products affected, information about patches for de-supported or earlier product releases, and how Oracle handles requests for additional information about security vulnerabilities.
- **Level 2:** *Critical Patch Update Availability Information for Oracle Server and Middleware Products* contains information related to the Oracle Server Technology product environment settings, minimum requirements, and patch availability. Oracle Server Technology products include Oracle Database, Oracle Application Server, Oracle Collaboration Suite, and Oracle Enterprise Manager Grid Control.

E-Business Suite Critical Patch Update Note contains patch information related to the Oracle Database, Oracle HTTP Server, Oracle Developer Suite, Oracle JInitiator, and Oracle E-Business Suite that are applicable to the E-Business Suite environment. It also includes known issues.

- **Level 3:** Bundled Patch READMEs accompany individual patches, and include instructions to apply the specific patch.

Prior to CPUJul2006, the Level 2 documentation was called Pre-installation Notes that listed the patches applicable to the Oracle environments.

CPU IMPLEMENTATION PHASES

The CPU implementation process consists of three phases:

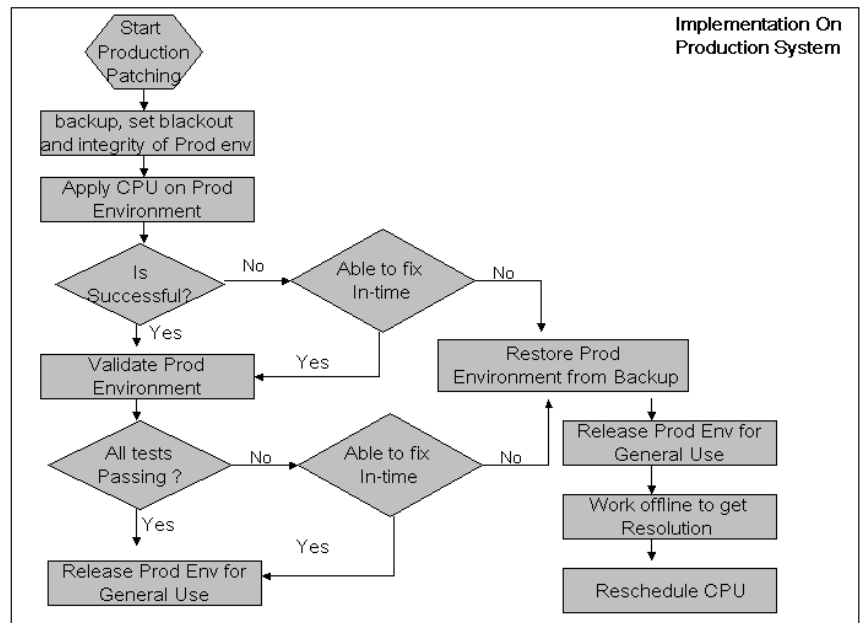
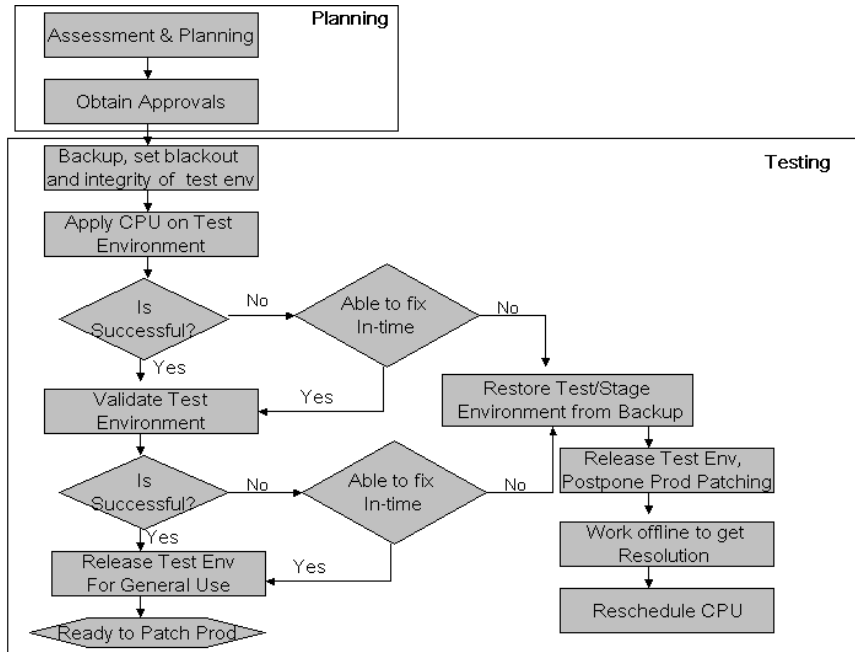
- Planning
- Test deployment
- Production deployment

The following flow diagrams show the main steps in each of the phases. Subsequent sections describe these steps and phases.

The high-level CPU implementation process can be broken down into three phases:

- Planning
- Test deployment
- Production deployment

Critical Patch Update Implementation Best Practices



Planning

Implementing a CPU in your IT environment is similar to implementing any other set of patches. To ensure a successful implementation, it is key to have proper change management systems and processes in place.

To ensure a successful CPU implementation, you need to:

- **Have proper change management systems and processes in place.**
- **Discuss and plan the CPU implementation with key participants.**

Assessment

During the initial planning phase, you should assess the criticality of the security flaws addressed in the CPU and communicate their criticality to management and other concerned parties in order to obtain the necessary approval. In addition, it is important to prioritize which systems will be patched first. The following guidelines should be followed:

- Have a complete inventory of Oracle products across your IT enterprise, with full version numbers, at your fingertips.
- Review the CPU Advisory to assess the criticality of the patch and extract an executive summary to obtain necessary approvals. Note that for each vulnerability newly fixed in the CPU, Oracle provides values for CVSS metrics indicating the preconditions required to exploit the vulnerability and the ease of exploit; and the impact of a successful attack in terms of confidentiality, integrity and availability to the targeted system. [MetaLink Note 394487.1](#) provides a detailed explanation on how the CVSS ratings are applied in the CPU documentation.
- Review the CPU Availability or Update Notes to identify the environments or systems that need to be patched and any pre-requisite patches that must be applied. In addition, refer to Known Issues documents for a given product for updates, issues and solution/workarounds.

Planning

As with any change, it is important to discuss and plan the CPU implementation with key participants. This ensures smooth execution. It is also important to assess risk and build contingency plans. The following guidelines should ensure the same:

- Download the required patches from Oracle *MetaLink* to a central storage system that can be made read-only accessible to target systems for CPU patching.
- Have a complete backup of all environments and a contingency plan to efficiently restore the system within the allocated downtime in case of unexpected failures.
- Ensure that the test and stage environments resemble the production systems as closely as possible, especially in product levels including patch sets and one-off patches. This tests the CPU effectively in the non-production environment first.

- Identify the key functions or areas in every system that should be tested after CPU implementation. A complete test plan or checklist of items to be tested would help here.
- Ensure that there is, at minimum, a one-day gap between test and production deployments. This allows you time to react to any unforeseen issues before moving on to production.
- Obtain approvals ahead of time for the downtime required for CPU implementation.

Approvals

It is imperative to pick a downtime with the least impact on your business and user base.

Obtaining approvals ahead of time for the downtime required for a CPU implementation is an important step of planning. Oracle provides an executive summary with a high level synopsis of the security defects in each product addressed by the CPU. This executive summary provides a "plain English" explanation of the vulnerabilities addressed in the CPU and can be used to brief management to obtain initial approval. Ensure that all affected members of the user community are aware of the planned downtime well in advance. It is imperative to pick the downtime with the least impact on your business and user base. For example, you may want to patch the test or stage environment on weekdays and to schedule downtime for production instances over the weekend.

CPUs are released every quarter. It will pay off to establish proper assessment, review and approval processes, if they do not already exist as a part of your change management system.

Test Deployment

When testing the CPU:

- Identify the areas where the patch needs to be applied.
- Ensure that test systems resemble production systems as closely as possible.

Oracle recommends that you test each CPU thoroughly on the test deployment before applying it to any production systems. For larger organizations with many production applications and systems, the initial phases of production rollout can be considered an additional component of the testing process. However, this should not replace testing in the test environment.

For effective production-readiness testing, ensure that the test environment resembles the production environment as closely as possible. CPUs depend on the patch set levels, family packs and one-off patches applied to an environment. Ensure that those patch set levels, family packs and one-off patches applied to the test environment match those applied to the production environment. If necessary, refresh the test environment from the production environment before testing the CPU.

Oracle patches use the Oracle Universal Installer (OUI) inventory to detect conflicts and to populate future records. Thus Oracle patches depend to a large extent on the accuracy and integrity of the OUI inventory.

Before applying a CPU to a test environment, Oracle recommends running some quick tests to ensure that the environment functions as expected.

Before applying a CPU to a test environment, Oracle recommends running required functional tests to ensure that the environment operates as expected. This helps isolate any pre-existing issues or issues that the CPU introduces.

If you encounter any conflict while applying the CPU, and if that conflict is not documented as ignorable, then you must stop patching and report the conflict to Oracle Support.

If there are no errors in patching, verify that the CPU patch has been applied successfully by checking OraInventory through the `opatch` command for Oracle technology products and check `ad_bugs` or `ad_applied_patches` tables for Oracle E-Business Suite patches. After this you also must run the required tests to ensure environment functions as before. This post-deployment validation task is equally important to avoid any surprises or risking production environment later.

Production Deployment

A few general tips for implementing a CPU:

- Ensure that planned downtime announcement page covers all the target systems before you shut down systems for patching. The downtime required may vary from CPU to CPU and from environment to environment. If you have large number of production systems to update, be prepared to refine the time estimate as you learn more about a given CPU.
- Ensure that you have a complete backup of the production environment and a restore strategy as a contingency plan for any unexpected failures.
- Verify the accuracy and integrity of the OUI Inventory through `opatch` and/or OUI commands. If you choose to ignore the OUI inventory during CPU patching, patch conflicts cannot be detected and you may overwrite previous fixes inadvertently. For more information about OUI Inventory, see the [Understanding and Automating Operations around Oracle Installer Inventory](#) white paper.
- You can detect potential patch conflict in advance by running the following command:

```
opatch apply -silent -no_bug_superset -report
```

The `-report` option detects any conflicts without applying patch.

- When you apply a patch using `opatch`, it rolls back any subset patches that are present in the environment. By default after every subset patch is rolled back, it re-links dependent binaries or libraries. For example, if a given patch supersedes five patches, then re-linking occurs five times during the

If you choose to ignore OUI inventory during patching, patch conflicts cannot be detected and you may overwrite previous fixes inadvertently.

Critical Patch Update Implementation Best Practices

rollback action. This re-linking action takes additional time. However `opatch` provides a `-norelink` option that avoids re-linking during rollback. If you use this option, you need to re-link manually using the `makefile` command provided with the patch after `opatch` completes. For example, for patch 4560421, the commands are:

```
$> cd $ORACLE_HOME/.patch_storage/4560421  
  
$> sh 4560421_make.txt | tee 4560421_relink.log
```

- Oracle E-Business Suite Environments are patched using the `adpatch` utility. Following are few `adpatch` options that you may want to consider while patching as they might help reduce time:

The ***noautoconfig*** option skips Autoconfig after patching. If you are applying multiple patches for Critical Patch Update, you may want to skip running Autoconfig for all the patches but the last one.

The ***nocompilejsp*** option skips JSP compilation during patching, allowing you to defer JSP compilation until pages are accessed, to reduce down time.

The ***nomaintainmrc*** option skips Multiple Reporting Currency (MRC) updates during patching. If you have MRC configured in the environment, you can skip update it once after all patching is completed using the `adadmin` utility.

- Before shutting down the production environment, run the required tests or health checks to verify the integrity and health of the production environment.
- Copy required patches from a central storage location to a local stage area on the target server(s). As patches may write to the patch stage area during patch application, the Oracle user or group must have write permission to the local patch stage area.
- Carefully review the appropriate CPU Availability or Update Notes to understand the dependency and sequence among patches.
- Follow the instructions for patch application, including any pre-implementation or post-implementation steps required, in the patch README.
- Scan the patch output logs and ensure there are no errors.
- If there are any errors or failures, then look at Known Issues document for the product on *Oracle MetaLink* to check if it's a known issue and apply the solution/workaround given in the document.
- Follow the patch verifications steps (where provided) and ensure the patch is applied successfully.

- Once a CPU is applied successfully, verify that the environment is functional. Follow through the checklist and functional-tests identified earlier and run them thoroughly in the patched environment.

The following tools and strategies can help to optimize your CPU implementation:

- **Environment Checker Tool**
- **Testing strategy**
- **Cook Book**
- **Merged application patches**
- **Shared product homes**
- **Oracle Enterprise Manager**
- **Proactive maintenance**

OPTIMIZING CPU IMPLEMENTATION

Oracle minimizes the cost and risk of implementing CPUs, but you may be able to further optimize CPU implementation by using processes and tools customized for your IT needs.

The following sections contain suggestions (in no particular order) that may help reduce the cost of CPU implementation in terms of both labor and downtime. Where you have large numbers of Oracle systems to update, the benefits resulting from these suggestions multiply.

Environment Checker Tool

The integrity of the target environment is critical for a successful CPU implementation. Otherwise you may end up debugging issues during CPU implementation that are not related to the CPU. These guidelines will help to verify the integrity of environment:

- Check OUI Inventory for patches using the `opatch lsinventory` command.
- For the Oracle E-Business Suite instance, ensure that any customized configurations are auto-configuration enabled. You can use the `adchkcfg.sh` command to identify configuration customizations in your environment and baseline them by creating appropriate template files in a custom directory. This prevents customizations being overwritten during patching, because certain patches might invoke auto-configuration by default.
- Before applying the CPU, obtain list of any invalid objects in the database. After implementing the CPU you can then check if the CPU has introduced any invalid objects.
- Run the required functional tests to ensure the environment is fully functional before you bring it down for CPU implementation.

Automating any of these checks helps reduce cost and human error, and can also increase the success rate of CPU implementation.

Testing Strategy

This is one of the most critical and challenging steps for an effective change management system. Unless explicitly documented, CPUs do not change features or functionality, or alter product APIs. Implementing a CPU has minimal, if any, impact on product behavior, user experience and any custom applications that you

Each CPU has minimal, if any, impact on product behavior, user experience and any custom applications that you have.

Thus you can focus on productive testing that gives maximum yield to your investment.

have. Thus you can concentrate on productive testing that gives maximum yield to your investment. Identifying the right set of tests that ensures the health of the system and automating them where possible increases the efficiency of CPU implementation.

Cookbook

Implementing a CPU usually involves reading the accompanying documentation, including CPU Availability or Update Notes and READMEs for individual patches. This documentation is designed for a general audience, and is likely to include sections that may not be applicable to your IT environment.

You can identify those portions of the documentation that are applicable to your environment and use them to create a cookbook during initial CPU analysis and testing. Also you may have certain pre-patch steps, such as shutting down systems in a specific order, and post-patch steps, such as functional validation, and so on that are unique to your IT environment. You can insert these and any other custom steps in this cookbook. This is particularly beneficial if you have large numbers of identical systems to update with the CPU. Having a cookbook not only reduces the time and effort required to implement a CPU but can also increase the reliability and predictability of CPU implementation.

For more information about the Ad merge patch utility, see the [Oracle Applications Maintenance Utilities](#).

Merge Application Patches

Use the Ad Mrg Patch utility (`admgrpch.sh`) to merge multiple Oracle E-Business Suite patches into a single bundle. This is highly recommended if you have multiple E-Business Suite environments with identical versions. This helps to reduce the downtime and effort required to apply the patch.

Shared Product Homes

Where possible, consider using shared product homes for multi-node deployments. For example sharing the Oracle Applications product home `APPL_TOP` across multiple identical application middle tiers for a given environment could significantly reduce the patching time, as the application patches need to be applied only once. Recently Oracle has introduced support for a shared application file system that includes sharing of `APPL_TOP`, `iAS ORACLE_HOME` and `Developer6i ORACLE_HOMES`. This will greatly reduce the efforts required to implement a CPU in multiple middle-tier environments.

Shared product homes are more of an architectural decision than patching so thorough evaluation needs to be done before adopting this architecture.

Choose The Right Tools

Each IT organization has unique needs so there is no single tool that fits everybody's need. However it is important to leverage an existing management infrastructure and change management tools for CPU implementation. For

For more information about Oracle Enterprise Manager (EM), see the EM Web site at: http://www.oracle.com/enterprise_manager/index.html

example, Oracle Enterprise Manager (EM) provides a central management console to manage Oracle environments in your enterprise. Using this management console, you can:

- Announce planned downtimes across multiple hosts
- Identify patches applicable to your environment using EM-rich reporting capabilities to find out what is applied and what is not.
- Schedule CPU patching across multiple hosts using a job system. Apply patches to multiple ORACLE_HOMEs at the same time (scalable patching model).
- Take corrective action and use the retry capability, where the job fails.

You may also consider automating parts of the CPU implementation tasks by developing the necessary extensions to existing change management tools or systems.

Proactive Maintenance

Generally, Oracle release CPU patches for the last two patch set releases only. Proactively maintaining systems by applying the latest Oracle patch sets and family packs at the right time helps avoid other upgrades during CPU implementation.

CONCLUSION

A CPU is similar to any other set of patches that you apply on a regular basis except that, for many, a CPU implementation is proactive, rather than reactive, maintenance. The appropriate change management systems with the right set of technology and processes make implementing a CPU a routine maintenance job.

Oracle On Demand successfully implements CPUs for hundreds of On Demand customers using these best practices, shortly after the CPUs are released. The Oracle On Demand business has reduced costs and downtime customers by successfully implementing many of the recommendations listed in this document.

Critical Patch Update Implementation Best Practices



CPU Implementation Best Practices

December 2006

Author: Rajesh Shah

Contributing Authors: Eric Maruice, Gopal Parthasarathay Ajay Srivastava, Niloy Banerjee, Darius Wiles, Debashis Saha, Sudip Datta

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

Copyright © 2006, Oracle. 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, JD Edwards, and PeopleSoft, are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.