

Oracle Identity Manager: Upgrading to 11gR1
White Paper
April 2011

Contents

1 Introduction	4
1.1 <i>Benefits of Upgrading</i>	4
1.2 <i>Pre-Requisite Reading</i>	4
2 Upgrade Impact	5
2.1 <i>Application Server and JVM Support</i>	5
2.2 <i>Request Management</i>	5
2.3 <i>Authorization</i>	6
2.4 <i>Access Policies</i>	7
2.5 <i>Scheduled Tasks</i>	7
2.6 <i>UI Customization</i>	7
2.7 <i>Entity Adapters and Event Handlers</i>	7
2.8 <i>APIs and Client applications</i>	8
2.9 <i>Connectors</i>	9
2.10 <i>Single Signon Integration</i>	9
3 Planning the Upgrade	10
3.1 <i>Upgrade Assessment</i>	10
3.2 <i>Upgrade Infrastructure</i>	10
3.3 <i>Pre-Upgrade Checklist</i>	11
3.4 <i>Pre-Upgrade Utility</i>	11
4 Upgrading Oracle Identity Manager	12
4.1 <i>Introduction</i>	12
4.2 <i>Upgrading the Development Environment</i>	12
4.3 <i>Upgrading the Test Environment</i>	12
4.4 <i>Upgrading the Production Environment</i>	13
4.5 <i>Upgrade Process Overview</i>	14
4.6 <i>Install additional Schemas</i>	15
4.7 <i>Install and Configure WebLogic Application Server</i>	15
4.8 <i>Install OIM 11gR1</i>	15
4.9 <i>Database-tier Upgrade</i>	15
4.10 <i>Middle-tier Upgrade</i>	15
5 Upgrade Best Practices	16
5.1 <i>General Recommendations</i>	16
5.2 <i>Project Initiation Considerations</i>	18

<i>5.3 Preparing Your Technical Environment</i>	19
<i>5.4 Upgrading Your Data</i>	21
<i>5.5 Training</i>	22
<i>5.6 Post-Upgrade Activities</i>	22
6 Conclusion	24

1 Introduction

Oracle Identity Manager (OIM) is an industry-leading identity administration and user provisioning solution. It provides operational and business efficiency through centralized administration and automation of identity and user provisioning events across the enterprise.

This white paper describes upgrading of OIM 9.1 deployments on Oracle databases to OIM 11gR1 PS1. For simplicity, we will refer to OIM 11gR1 PS1 as OIM 11gR1. The following OIM 9.1 versions with supported Bundled Patches (BP) are eligible for upgrade to OIM 11gR1:

- OIM 9.1.0.1
- OIM 9.1.0.2

For more details about the certification matrix refer to [System Requirements and Supported Platforms for Oracle Identity and Access Management 11gR1](#).

1.1 Benefits of Upgrading

OIM 11gR1 has been architecturally optimized for distributed, cloud, and in-house environments. Oracle continues to evolve the OIM product to offer our customers maximum return on their software investment. Here are some of the benefits that may be realized by an upgrade to OIM 11gR1:

- Provides access to new functionality and software applications that can help keep your organization well positioned to meet your business objectives through leveraging the latest technology and built-in business processes
- Provides an out-of-the-box solution that can be used as is or further customized to meet specific customer needs
- Provides compliance at a lower cost through retiring customizations
- Increases efficiency of applications and businesses by incorporating the latest performance and usability enhancements
- Provides a platform for the highest levels of product support

Before upgrading OIM 9.1 to OIM 11gR1, it is imperative to fully understand the functional differences, upgrade process, upgrade path and most importantly the criteria for considering an upgrade. For a complete list of new features in OIM 11gR1 refer to [OIM 11gR1 User's Guide](#).

1.2 Pre-Requisite Reading

Here are pre-requisite reading materials that will help you understand the OIM Upgrade:

- [OIM Architecture White Paper](#)
- [Oracle Fusion Middleware Installation Guide for Oracle Identity Management](#)
- [SOA Documentation](#)

2 Upgrade Impact

2.1 Application Server and JVM Support

Oracle Identity Manager 11g R1 supports only the WebLogic Application Server as the deployment platform. Support for WebSphere will be available at a later date. Application Servers such as OC4J and JBoss are not supported. Please refer to the table below to understand the Application Server and JVM combinations supported for Oracle Identity Manager 11g.

From AS on JDK combination (OIM 9.1.x)	To AS on JDK combination(OIM 11.1.1.5)
WLS on Sun or JRockit or HP JDK	WLS on Sun or JRockit
JBoss on Sun or JRockit or HP JDK	WLS on Sun or JRockit
WAS on IBM JDK	WAS 7.x on IBM JDK*
OC4J on Sun JDK	WLS on Sun or JRockit

Table 1: Application Server and JVM Support Matrix

* Support for IBM WebSphere will be made available shortly after the release of Oracle Identity Manager 11gR1 PS1 (11.1.1.5)

2.2 Request Management

The key functional differences between OIM 9.1 and OIM 11gR1 releases includes

- SOA Composite based Approval Processes
- Support for new types of Requests like Create User, Assign Roles, etc.
- Enhanced Bulk Requests
- Request Templates
- Request validation prior to submitting the request

For more details about Request related features please refer to the [OIM 11gR1 User's Guide](#).

2.2.1 Comparison of request-related objects

From a migration perspective, the following table maps the OIM 9.1 to OIM 11gR1 request objects:

OIM 9.1	OIM 11gR1	Migrated
Approval Processes	SOA Composites	Yes
Process Determination Rules	Approval Policies	Yes
Object Forms	Request Datasets	Yes
Pre-Populate Adapters	Pre-populate Plugins	No
Email Definition	SOA Notification	No
Task Assignment Adapters	SOA Human Task routing rules	No

Table 2: Object Differences

2.2.2 Approval Process Migration

The migration process will convert most Approval Processes to SOA composites. However, some Approval Processes may be partially migrated or not migrated at all, under the following conditions:

- Approval Processes associated with Organization provisioning are not upgraded as part of mid tier upgrade. Request-based Organization Provisioning is not supported in OIM 11gR1.
- Approval Processes are partially upgraded if any of the following applies:
 - Event handlers or adapters are associated with “System Validation” and “Provide Information” tasks.
 - Task information is used in creating the task assignment rule.
 - Process information is used in framing the task assignment rule with attributes Process Name, Process Type.
 - Task of the approval process is associated with Task Assignment Adapter or Event Handler or Process Task Adapter.

2.3 Authorization

Application access controls to perform various operations are controlled by the authorization engine embedded in OIM 11gR1 with the help of authorization policies. The authorization policies determine at runtime whether a user is allowed to perform a particular action or not. You can define authorization policies that satisfy the authorization requirements within OIM 11gR1.

In OIM 9.1, each component defined and managed its own authorization policy UI and implementation.

In OIM 11gR1, authorization policy management is centralized as an administrative feature and enforced by an embedded version of Oracle Entitlements Server (OES), Oracle's industry-leading fine-grained entitlements administration product. These authorization policies secure access control to the OIM 11gR1 application, thereby defining "who can do what" inside the application.

For more details about OES refer to:

http://www.oracle.com/technology/products/id_mgmt/oes/index.html

The following functional areas are protected by the new OES authorization policies:

- Event management APIs
- Reconciliation APIs
- Reconciliation Event Management UI
- Identity Administration (User and Role)

2.4 Access Policies

Access Policies have been enhanced significantly to support improved automated provisioning of multiple accounts in the same instance of target application to the same user, as well as automated provisioning of multiple accounts in different instances of the same target application. The overall goal is to reduce the need for cloning of objects and improve performance.

From a migration perspective, all Access Policies are migrated as-is.

2.5 Scheduled Tasks

All scheduled tasks are migrated automatically to the OIM 11gR1 model of Scheduled Tasks and Jobs. For more details refer to the [OIM 11gR1 Administrator's Guide](#). The content of the OIM 9.1 Job History are obsolete after the Upgrade to OIM 11gR1.

2.6 UI Customization

The OIM 11gR1 upgrade process does not retain some UI customizations done in OIM 9.1 environment. UI customizations, if any, must be revised. For more information on UI Customizations, please refer to the [OIM 11gR1 Developer's Guide](#).

2.7 Entity Adapters and Event Handlers

2.7.1 Entity Adapters

Entity Adapters are used to customize operations on entities like User and Role. They are frequently used to populate attributes for entities like User and Role at various lifecycle events like pre-update, pre-delete, pre-insert, post-insert, post-update, or post-delete.

In OIM 9.1 customizing the behavior of life cycle management events like create, modify, enable, disable, etc., on User and Role entities were based on the Dataobject model. In OIM 11gR1 this has been updated and is based on the Orchestration model. To pre-populate an attribute for User life cycle management events, an event handler must be written to populate the attributes programmatically. The table depicts the mapping between OIM 9.1 and OIM 11gR1 user entities

9.1 Entity Adapter		OIM 11gR1 Event Handler		
Form Name	Handler	Entity Type	Stage	Action
User	Pre Insert	User	Pre Processing	CREATE
User	Post Insert	User	Post Processing	CREATE
User	Pre Update	User	Pre Processing	MODIFY, DISABLE, ENABLE, LOCK, UNLOCK
User	Post Update	User	Post Processing	MODIFY, DISABLE, ENABLE, LOCK, UNLOCK

User	Pre Delete	User	Pre Processing	DELETE
User	Post Delete	User	Post Processing	DELETE

Table 3: Entity Adapters Migration

2.7.2 Event Handlers

In 9.1, Event Handlers provided the capability to add various customizations on a certain Data Object life cycle events like Insert/Update/Delete via Data Object Manager in Design Console.

With 11g, User Management, Role Management and Request have been re-architected to use the Orchestration Framework. Customers can define various event handlers for life cycle management events like Create/Update/Delete. The Orchestration Framework also provides the capability for event handlers to be executed in either a synchronous or asynchronous fashion. The following table depicts the mapping for Role entity. Similar mapping will also be applicable for User and Request entities.

9.1 Event Handlers		OIM 11gR1 Event Handler		
Form Name	Handler	Entity Type	Stage	Action
User Group	Pre Insert	Role	Pre Processing	CREATE
User Group	Post Insert	Role	Post Processing	CREATE
User Group	Pre Update	Role	Pre Processing	MODIFY
User Group	Post Update	Role	Post Processing	MODIFY
User Group	Pre Delete	Role	Pre Processing	DELETE
User Group	Post Delete	Role	Post Processing	DELETE

Table 4: Role Entity Adapters' Migration

2.8 APIs and Client applications

2.8.1 Signature based Login

In OIM 9.1, customers used signature-based login to connect remotely to OIM and execute OIM APIs. OIM 11gR1 continues to provide support for this mechanism. However, Oracle recommends that customers should plan on using the username-password approach of logging into to OIM. To do this in a secure manner, they should use the Credential Store Framework. For more details refer to the [Application Security Guide](#)

2.8.2 APIs

Oracle provides a network-aware, Java-based application programming interface (API) that exposes services available in Oracle Identity Manager. This API is based on Plain Old Java Objects (POJO) and takes care of all the plumbing required to interact with Oracle Identity Manager. This API can be used for building clients for Oracle Identity Manager and for integrating third-party products with the Oracle Identity Manager platform.

In Oracle Identity Manager OIM 11gR1 Release, some of the legacy APIs have been rewritten to use the new architecture. The corresponding utility services or interface classes have been changed. The following table provides a high-level mapping between the legacy and new interfaces.

Legacy Service	New Service
Thor.API.Operations.tcUserOperationsIntf	oracle.iam.identity.usermgmt.api.UserManager
Thor.API.Operations.tcGroupOperationsIntf	oracle.iam.identity.rolmgmt.api.RoleManager
Thor.API.Operations.tcOrganizationOperationsIntf	oracle.iam.identity.orgmgmt.api.OrganizationManager
Thor.API.Operations.tcRequestOperationsIntf	oracle.iam.request.api.RequestService
Thor.API.Operations.tcSchedulerOperationsIntf	oracle.iam.scheduler.api.SchedulerService
Thor.API.Operations.tcEmailOperationsIntf	oracle.iam.notification.api.NotificationService

Table 5: API Differences

2.9 Connectors

Connectors are automatically upgraded as part of the server upgrade. To use newer versions of the connector, however, customers must be upgraded to the versions supported by Oracle 11gR1. They can make use of the Connector Lifecycle Management feature that automates Connector upgrades.

2.10 Single Signon Integration

Oracle Identity Manager 11g has adopted the industry standard model of J2EE security for Single Signon. This allows customers to implement custom authentication mechanism, implement standards-based security and follow security best practices. As a result of this, the HTTP Header variable-based approach that was supported in 9.x has been deprecated. Customers who use CA Siteminder or Tivoli Access Manager must deploy the respective J2EE Agents for Weblogic. For instructions on configuring Oracle Identity Manager 11g with Siteminder or Tivoli Access Manager, customers should contact Oracle Support at <http://support.oracle.com>

3 Planning the Upgrade

An upgrade project is similar to an implementation project; however, upgrade projects can be significantly more efficient than implementations because they leverage your previous implementation efforts and outputs. In addition, upgrades can be executed within the current change-management system used by your organization. To ensure a quick and successful Upgrade, Oracle offers specialized expertise to support customers and partners performing upgrades and migrations.

In most instances, the OIM 11gR1 upgrade process will be carried out in the following sequence:

3.1 Upgrade Assessment

Because OIM 11gR1 offers greatly enhanced functionality, better performance, and a redesigned UI, upgrade projects must address human and functional impacts as well as technical considerations. For this reason, the recommended upgrade approach includes performing an Upgrade Assessment prior to initiating the actual upgrade process.

The Upgrade Assessment should achieve the following objectives:

- Assess the current OIM environment including Connectors
- Analyze new product functionality. Review the Release notes and identify work arounds if any
- Evaluate the complexity of the upgrade
- Estimate the level of effort to upgrade
- Assess business processes, existing customization, functional redesign, and current requirements versus obsolete requirements that may be out of scope
- Recommend upgrade team roles
- Review interface migration tasks
- Plan for upgrade tuning
- Identify data migration tasks
- Provide logical end-user training recommendations

Based on a comprehensive analysis of the current deployment, the Upgrade Assessment provides valuable input into subsequent steps as well as a detailed rendering of the level of effort involved to complete the upgrade.

3.2 Upgrade Infrastructure

The first phase of the actual project is to upgrade your hardware and software to meet system and implementation requirements.

Hardware sizing should be planned based on the combination of enhanced OIM product functionality, technological change, anticipated changes in the way you use the applications, deployment of additional users or server processes could all impact sizing requirements for the upgraded solution. Performance and load testing can help determine if the hardware is adequate to support your production requirements. For more details refer to the [Oracle Identity Manager Sizing Guide](#).

Software including Operating System, Middleware and Database should be in accordance with OIM 11gR1 certification matrix. For more details about the certification matrix refer to [System Requirements and Supported Platforms for Oracle Identity and Access Management 11gR1](#).

3.3 Pre-Upgrade Checklist

These tasks must be completed before you start the upgrade:

- It is very important to take backups of the OIM schema before you perform upgrade. This will ensure that we have a well defined recovery path in case of any failures during Data Tier upgrade
- Read the upcoming release notes
- Understand the order in which you will upgrade OIM 9.1 components and document it. Share your upgrade plans with all key stakeholders

3.4 Pre-Upgrade Utility

Run the Pre-upgrade utility, *generatePendingTransactionReport.sh*, before performing the upgrade. The Pre-upgrade utility will generate a report that will list the following:

- Pending Recon: All events not in Event Closed, Event Linked state. This will include all events that are deferred to a later date
- Pending Request: All requests (including self registration) pending approval
- JMS Queues: List all the messages, which are in unprocessed state. A JMS message is submitted in case of off-line provisioning. So this report will provide details of the entire pending provisioning task including task, which are initiated in offline mode
- Audit: In OIM 9.1, any pending changes (in AUD_JMS table) of entities being audited need to be processed prior to upgrade. The row count of aud_jms table should be zero
- Based on this summary report take necessary actions before Upgrade

Finish the pending transactions in 9.1 environments before starting the upgrade. Following is the list of Pending transactions which can exist in customer environment:

- All pending recon events generated in OIM 9.1 must be processed. Failed events will not be upgraded, if left unaddressed
- All pending approval tasks generated in OIM 9.1 must be processed. Pending approval tasks will not be upgraded, if left unaddressed. Ensure that these reach a final state before the Upgrade.
- Scheduled task execution history will be left in OIM 9.1 and will not be Upgraded – ensure that all pending scheduled tasks have finished and no scheduled task run is underway during the Upgrade period
- Any asynchronous audit processing must be processed in OIM 9.1 prior to upgrade
- All JMS queues should be empty. JMS messages in regular or error queues will not be migrated in OIM 11gR1. Wait for all JMS messages to be consumed, take proper manual actions for JMS error messages and clean all JMS tables before upgrading to OIM 11gR1.

4 Upgrading Oracle Identity Manager

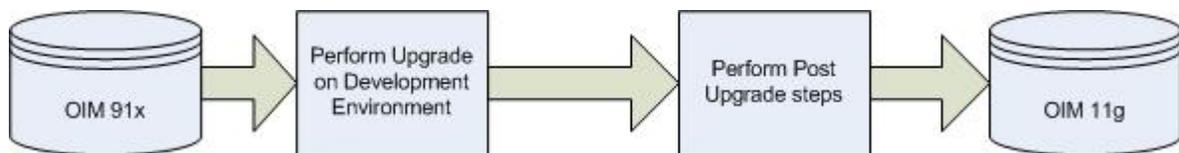
4.1 Introduction

Before you perform the steps involving upgrade, make sure you have completed all the pre-upgrade tasks and checklists. Oracle recommends performing the Upgrade in a Development and Test environment before upgrading the Production environment. This will ensure that upgrade is of high quality, risk free and minimum downtime.

The Upgrade Assistant (UA) utility is used to upgrade the Database-tier, Middle-tier and report the outcome of the upgrade process

4.2 Upgrading the Development Environment

Upgrade should first be performed in development environment. Following diagram outlines the upgrade process in development environment.



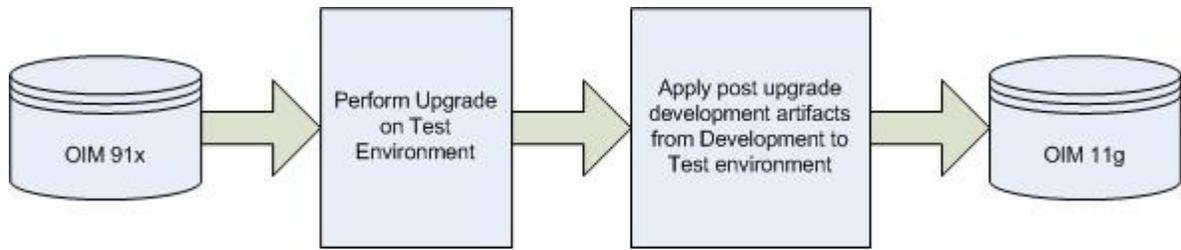
The overall upgrade process involves post upgrade steps which can potentially involve significant amount of development work. Customers can also choose to make additional customizations on top of existing environment leveraging advanced OIM 11gR1 features.

The development artifacts and metadata created as part of post upgrade steps can directly be reused as part of Test and Production upgrade using OIM 11gR1 Test to Production (T2P) guidelines. Document any changes to the Upgrade process and follow them in the Test and Production environments. Adhering to these processes will provide high quality, risk free and minimum downtime experience during upgrade. For more details refer to [OIM 11gR1 System Administrator's Guide](#)

4.3 Upgrading the Test Environment

It is extremely important that the upgraded release be tested to validate its function and performance before deploying it to users. In addition, the upgrade process must be tuned to minimize the time required to perform the production upgrade and optimize the environment for improved performance at deployment.

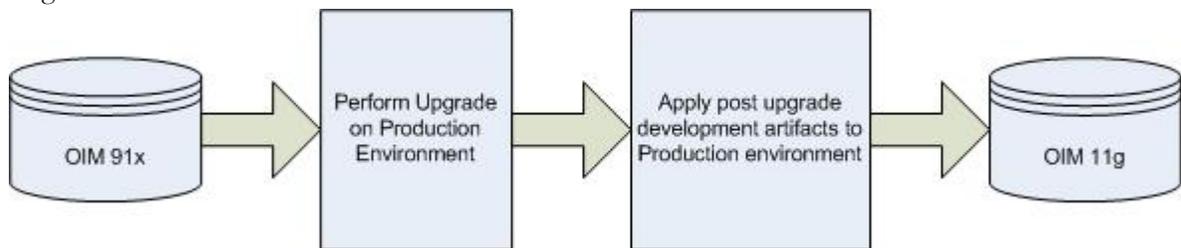
Following figure depicts outline of upgrade in Test environment.



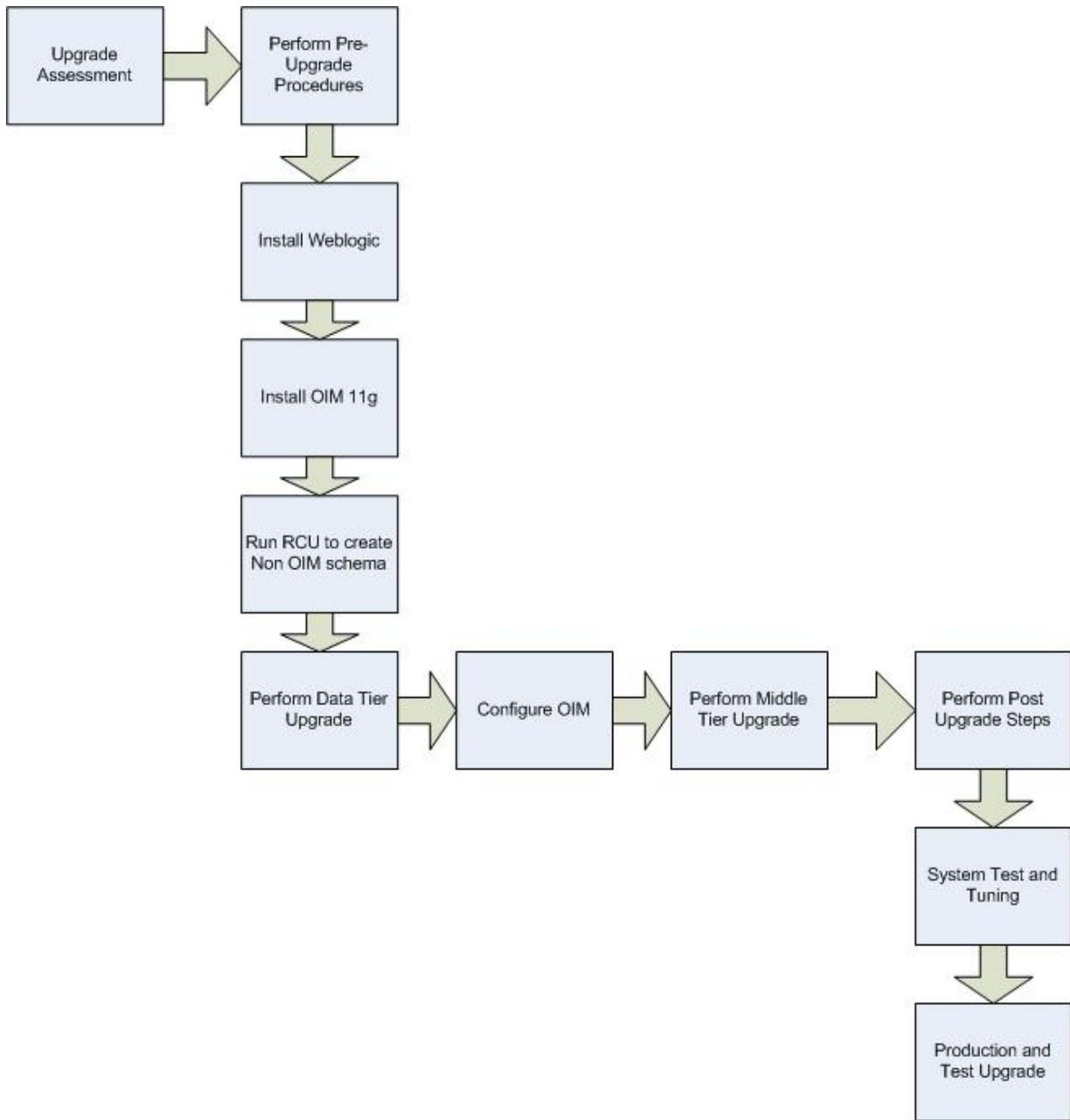
In test environment, Upgrade must be performed using the Upgrade process as we discussed above. Once upgrade is complete, we can reuse the development artifacts created during post upgrade step in development environment and apply them in test environment using OIM 11gR1 T2P guidelines. This way we reduce the time to upgrade and also verify various artifacts generated during upgrade.

4.4 Upgrading the Production Environment

The production environment is your live, deployed business instance where your internal and external users interact with the OIM and generate actual business data. Because of the “customer-facing” nature of the production environment, the upgrade process needs to very efficient, error free and fast. From Upgrade process perspective, the process will be similar to Test Environment Upgrade as outlined in diagram below.



Within each environment, the standard upgrade process flow is detailed in the graphic below. For detailed steps refer the upcoming OIM 11gR1 Upgrade Guide:



4.5 Upgrade Process Overview

OIM upgrade process supports an “In-Place” upgrade for Database-tier and “Out-of-Place” upgrade for Middle-tier upgrade. For “Out-of-Place” upgrade, OIM 11gR1 must be installed on a separate environment and the UA utility needs to be used to move all the data from their existing OIM environment to the new OIM 11gR1 environment.

Here is the summary of steps to upgrade from OIM 9.1 to OIM 11gR1:

- Install additional Schemas
- Install OIM 11gR1
- Database-tier Upgrade

- Middle-tier Upgrade
- Post Upgrade Tasks

For detailed steps refer the upcoming OIM 11gR1 Upgrade Guide

4.6 Install additional Schemas

The following additional schemas must be setup using the Repository Creation Utility (RCU):

- MDS schema used by OIM
- SOA schema used by BPEL

4.7 Install and Configure WebLogic Application Server

Install WebLogic Application server and configuration OIM 11gR1 on WebLogic Application Server. For more details refer to the [Installation Guide for Oracle WebLogic Server](#)

4.8 Install OIM 11gR1

This step will install all the binaries and create an OIM Home for OIM 11gR1

4.9 Database-tier Upgrade

Database-tier upgrade is an In-Place upgrade, which means upgrade is performed on 9.1 schemas. This is done using the UA utility's Metadata Repository (MR) Upgrade Plug-in. During this phase:

- Upgrade is performed by pointing the UA MR plug-in to a 9.1 schema.
- The UA MR plug-in carries out all the schema changes required in 9.1 schema to upgrade it into an OIM 11gR1 schema.

4.10 Middle-tier Upgrade

Middle Tier upgrade involves upgrading the OIM middle tier. This is an Out-of-Place upgrade performed via UA utility's Middle-tier plug-in, which in nutshell does the following:

- Upgrade is performed by pointing the UA to a 9.1 source topology as well as an OIM 11gR1 destination topology
- The destination topology must be created, using the standard OIM 11gR1 installation and configuration process, prior to the execution of the UA
- The UA MT plug-in carries over all of the required changes from the source environment to the destination

5 Upgrade Best Practices

In preparing to successfully upgrade, there are multiple considerations to include in your upgrade planning and execution efforts. These recommendations are intended to help you learn and manage a successful upgrade project.

5.1 General Recommendations

The following general considerations should form the backbone of your upgrade initiative.

5.1.1 Determine Your Upgrade Path

Verify whether you can upgrade directly to the latest release or whether you must first upgrade to a previous release before moving to this target release. In addition, evaluate the complexity of your upgrade effort based on number of customizations, number of integration points, and number of interfaces. Finally, determine the metrics and cost associated with each aspect of the upgrade. Each consideration should be addressed through a thorough upgrade assessment.

5.1.2 Treat Your Upgrade Activity as a Formal Company Project

The single best predictor of upgrade success may be the planning and project management rigor invested. A structured approach for managing the tasks, resolving issues, and measuring progress is absolutely critical. Equally important is a clearly defined and documented project scope. A defined scope is critical to project measurements necessary for time and cost containment. Experience has demonstrated that clear issue definition, strong project management, and executive ownership are critical success factors to a well-performing project effort.

5.1.3 Use Change Management Appropriate for an Upgrade

During an upgrade, it is imperative to freeze metadata and system data in your production environment. With respect to the new release, ensure all relevant patches available are applied appropriately. Failure to effectively manage these different change management requirements can result in upgrade step failures and unexpected user acceptance test results.

Once you have addressed this consideration, you should proactively search for issues throughout your upgrade effort and schedule relevant updates until you reach a “go/no-go” milestone. At this point, you should enforce a content freeze to stabilize the environment.

5.1.4 Build an Upgrade Team with Broad and Complementary Skills

Several different skill sets will be necessary to successfully upgrade your system. The following list details recommended roles that should be staffed within an upgrade project team. Note that a steering committee is critical to success. Creation of an active and interested steering committee is imperative because critical business decisions must be efficiently made and dealt with throughout the project. Furthermore, project failures are often traced to the lack of an effective governance body.

A typical upgrade team should include the following members:

- Steering committee
- Business owner of the application
- Application data owner
- Key user group representatives
- Dedicated project manager
- Technical functional lead
- OIM administrator
- Database administrator
- Technical change management owner/release coordinator
- Operating system administrator(s)
- Testers—both technical and functional
- Technical upgrade architect(s) and specialist(s)
- Organizational change management/training lead

5.1.5 Utilize Peer and Oracle Resources

Most organizations sponsor upgrade projects infrequently, so it is important to leverage the experiences of others as much as possible. Use Oracle expertise by going through the Documentation and Support channels.

5.1.6 Decide When to Change or Add Business Processes

In most cases, there is functionality in the release being evaluated that will help your business improve processes and automate tasks. This can be a small enhancement to business processes you are already using, or larger changes such as incorporating a new business flow. One critical decision for your upgrade project is whether you will implement the new functionality as part of the upgrade, or upgrade your current processes without change, and implement new functionality as a follow-on project.

Generally, implementing your current processes in a new system can be a way to mitigate risk in the upgrade project. However, your business realities may preclude this approach, especially if the updated processes native in the software can markedly improve operations. For example, the business may be driving to take advantage of new capabilities as quickly as possible, or it may be more appropriate to modify processes and engage in a coordinated training effort to increase user adoption of the new solution.

By carefully weighing the pros and cons of these approaches, you can choose the best strategy for your organization.

5.1.7 Get Current Product and Upgrade Information

Make sure you are using available Oracle resources to help you gather current information for your project, and work with Oracle Support for critical case management throughout your conversion timeframe. Oracle has increased its focus on assembling assets to help customers with upgrades, and leveraging these resources will increase your ability to upgrade smoothly.

Oracle maintains several resources to ensure that you obtain the most relevant information for your

upgrade initiative. In addition, upon product general availability, make sure you get the most current documentation available which is a good starting point for all types of upgrade information.

5.1.8 Escalate and Resolve Problems as Appropriate

Use Oracle Support if you believe application issues are being experienced. Train your first-line staff to log cases early and as completely as possible, providing the appropriate trace files and environment information, and highlighting business and technical milestone dates that will help in determining case prioritization. The Oracle Support and your Oracle account team can help in this area.

To ensure that your project progresses as smoothly as possible, Oracle encourages you to escalate Priority 1 (P1) issues as early as possible. These issues are typically on the critical path for your go live, and getting the right focus on their resolution early will give you the best chance of staying ahead of schedule. Depending on several factors—including your customization, issue complexity, and more—an escalated issue can take appreciable time to resolve with a fully tested solution, so the earlier the process is started, the better.

Even when you encounter noncritical (non-P1) issues, it is recommended that you log cases through Oracle Support. Cases logged in this way are often resolved faster than calls to the Support Center. This is due, in part, to the details you can provide online:

- Detailed description of the issue in your own words.
- Clearly articulated priority and sense of urgency.
- Trace files, screen shots, log files, and other relevant information to help the analyst move quickly on your issue.

5.2 Project Initiation Considerations

There are a number of things that you can do as your project begins—or even before it formally kicks off—that will ensure the project has a proper foundation and is well positioned for success.

5.2.1 Prepare the Organization

As you enter the initial stages of an upgrade project, engage the entire affected organization to help them prepare for the work ahead and the changes they will experience in their jobs. Obtain formal buy-in from the stakeholder organizations and kick off the project in a face-to-face meeting. Formality, visibility to upper management, and team building can be key aids to securing the cooperation and problem-solving help you will need as the project progresses.

These discussions should include both the business impact of the change and the associated change schedule. For example, secure agreement on all business blackout periods necessary for system changes.

In addition, efforts should be made to promote the upgrade benefits. Focus on building excitement through formal communications and informal dialogue. An upgrade will likely benefit the company and every end user!

5.2.2 Inventory Your System

All configuration elements of your enterprise system should be inventoried and the current configuration items (versions) should be copied and stored for technical change management control. Upgrading is analogous to moving; before starting, you have to make sure you identify where all your belongings are and that they are being handled appropriately. Early in your project, be diligent in gathering this information through composing and completing a preliminary upgrade questionnaire. During the upgrade assessment and planning, you should inventory your system as follows:

- UI (screens, views, and applets)
- Customizations, extensions, and modifications
- Localizations
- Workflow processes
- Interfaces, APIs, and integrations
- Reports
- Data models enhancements and custom tables
- Connectors
- Third-party products
- Hardware
- Software releases and patches, including operating system and database
- Obsolete requirements

5.2.3 Prepare a Go Live Checklist

Once you have completed the initial planning, you can create a checklist of criteria to guide the ultimate deployment of the upgraded solution. The planning activities should allow you to develop a robust checklist to assess appropriate “go/no-go” decision points. Creating this checklist as soon as possible is a good way to organize project goals, validate your plan, and identify your success criteria before the pressure is on to complete the project. This list should be reviewed periodically before go live to ensure progress is sufficient to complete in time.

5.2.4 Understand and Mitigate Project Risks

Early in the project, a risk analysis should be undertaken to determine project risks such as resource contention, other projects going live at the same time, and so on.

For risks that have a high probability of occurring and have a large impact, specific mitigation plans should be developed. These plans describe, in advance, what actions to take if the risk becomes a reality. The analysis and plans need to be reviewed on a regular basis throughout the project.

Look for key points of failure, especially in the area of resource loading for your technical and business specialists. If you lack bench strength in any particular areas, develop a plan to supplement and/or back up critical personnel.

5.3 Preparing Your Technical Environment

While many of the activities required for a successful upgrade project involve end users and net change for the applications, you must also manage the changes to your technical environment carefully.

5.3.1 Calculate New Hardware Sizing

Given the potential changes to your current system configuration, it is absolutely vital to get an accurate sizing for your new architecture. The combination of enhanced OIM product functionality, technological change, anticipated changes in the way you use the applications, deployment of additional users or server processes, and possible implementation of new functionality could all impact sizing requirements for the upgraded solution. Accurate sizing information will help you decide whether you can reuse current hardware, need to increase hardware resources, or should consider upgrading one or more of your servers. Similarly, sizing considerations are important whether or not you intend to upgrade in place (with potential reuse) or switch to a new hardware platform during the upgrade process. Performance and load testing can help determine if the hardware is adequate to support your production requirements.

Using production volumes and realistic user scenarios to assess the performance of the system on the current version versus the upgraded version will provide the project team critical information on hardware sizing, areas requiring tuning, and quantitative response times. Customers should seriously consider investing in a formal performance and benchmarking team to complete these tasks. In the upgrade scenario, the output from this team will be more accurate scalability and capacity requirements for the application versus a less structured, ad hoc approach.

5.3.2 Identify Custom Code and Scripting

Any custom code integrated with the OIM may be impacted during an upgrade. It is important to not only identify any custom code but also track the progress of any retrofit efforts during the project. You need to identify the code, who owns the code, and its status.

All interfaces, form customizations, and customized reports will require extensive testing to ensure that they have not been affected by changes to tables or APIs in the upgraded software. In some cases, customizations can be removed following an upgrade if new features and functionality satisfy the business requirements previously met with the custom code.

5.3.3 Database Maintenance

From a general database perspective, there are a few actions that can be completed to assist the upgrade project. To optimize the efficiency of the system as you upgrade, you should defragment and reorganize the database to the largest extent possible beforehand. Your database administrator should be able to use current database management tools to accomplish this goal.

Plan for database maintenance during and after the upgrade. This consideration is critical for larger implementations to address issues such as chaining, fragmentation, statistics, and index rebuilds. Database maintenance is frequently overlooked, but is vital to a successful upgrade.

5.3.4 Follow the Specific Platform Recommendations

Refer to OIM documentation for details on the specific requirements and supported platforms for the target OIM version.

5.4 Upgrading Your Data

Once the system is installed, you should turn your attention to data considerations. This step is a critical one in your upgrade success, and often the most time sensitive.

5.4.1 Identify Data Migration Tasks

After the upgrade, there are various post upgrade steps which needs to be performed. This will result in generation of various Development Artifacts like SOA composites, UI customizations etc OR metadata like request datasets.

After completing a development environment upgrade or any necessary reconfiguration and unit testing, you must migrate the configuration changes and certain types of data and files from the development environment to your test or production environment using T2P (Test to Production) guidelines of OIM. Identify and catalogue these needs early in the process to ensure that the upgrade effort is not compromised.

5.4.2 Test with Production Data

Converting your data accurately and efficiently depends on the quality and makeup of the data itself. Working with a current representative subset of your production data will give you valuable information about how the testing process should be structured as well as how long it will take to complete. Typically, your first conversion will be the longest and the most difficult.

When performing the development upgrade, it is important to leverage a representative subset of production data instead of an exact copy; this is because the development environment usually has less capacity in both memory and hard drive space than the test and production environments. Once the development database has been completed, the test upgrade should be done with the latest clone of production. It is important to use the full production data in your test environment. This will save a lot of issues when you redo the upgrade in Production. Any security concerns can be mitigated by using data masking technologies. For more details refer to [Oracle Data Masking](#).

5.4.3 Leverage Existing Test Scripts and Plans

To prepare test scripts for use during the testing cycles, begin with the test scripts utilized during the original implementation and augment these to include testing of the new features and functionality. Also consider any modifications needed for process flow changes related to the upgrade.

If these materials do not exist from the original implementation or previous upgrade, create them and store them in a library. This material can save significant time in preparing for upgrade testing.

5.4.4 Choose a Minimum Number of Test Conversions

An effective testing cycle will involve a minimum number of testing runs. Successful upgrade projects test the data conversion multiple times. Depending on the complexity, volume, and success of the process, you might need to practice fewer than five 5 times or more than 20. This practice instills confidence in the accuracy of the final conversion timeframe. Testing after your first successful conversion will help you prove repeatability in the process.

5.5 Training

While many of your team's existing skills will serve you well in your upgrade project, it is extremely important to provide training on the new features, business processes, and other changes, to help your staff become more efficient and effective with the new version.

5.5.1 Train End Users on the New Solution

When you first implement the system, your end users must be trained from the ground up to use the new application solution. However, during your upgrade, you will likely have experienced users who are already familiar with the basics. This fact can both assist and impede your upgrade project effort. Your end users, most importantly those who will be testing the system, must have good information about how the resulting solution will be different—that is, whether the changes are functional, technical, or interface-related in nature. These considerations will prevent issues from being reported that result from misunderstandings, and better position overall acceptance of the new solution.

5.5.2 Get Specific Technical Training

The project team and support team must be proficient in the new technologies introduced in the latest release. Team members must also understand the new architecture and performance best practices. An assessment is recommended to reconcile the skills needed to support the development and maintenance of the new release. This consideration is especially critical for a successful upgrade initiative. Some of the trainings include:

- Oracle Identity Manager 11g: Essentials (D65160GC10)
- Oracle Identity Manager 11g: Develop Identity Provisioning (D65156GC10)
- Oracle SOA Suite 11g: Build Composite Applications (D53946GC20)
- Oracle SOA Suite 11g: Administration (D64648GC10)
- Oracle SOA Suite 11g: Essential Concepts (D58786GC10)

5.5.3 Optimize Training Processes

One of the best ways to reduce the number of issues you have to track, research, and resolve is to train your users at an optimal time in the upgrade process. Although many of the core functional and administrative business processes are similar between releases, training will give your users the information they need to distinguish true issues from intended changes. For this reason, the timing of this functional training is important. While you may want key users to be trained early on to give input to the project and assess impact, most users prefer to be trained closer to going live on the new system so they don't have to remember what they have learned over a long period of time without being able to apply this knowledge.

5.6 Post-Upgrade Activities

Once the core technical upgrade has been completed, there are several additional steps to ensure success.

5.6.1 Secure Functional User Buy-In

Functional validation of the system is a key task. Most projects use functional users, away from their main responsibilities, to accomplish this objective. Though it may be self-evident, if you have functional users

complete testing, they must see the value of the process and share the project goals to complete the task effectively. Typically, these resources are setting aside important tasks to participate in the upgrade initiative, so take the time to solicit both management and individual cooperation. Once this cooperation is achieved, ensure you are collectively allocating enough time to complete a thorough testing cycle.

5.6.2 Testing Scope

A comprehensive testing effort is one of the key steps to finishing the upgrade and going live on the new release. As such, it is important that you consider the testing element of the upgrade as a major software update. Typically, a full, integrated test is performed that includes user acceptance and performance testing, and exercises all the business processes the organization will use. This testing regimen must be detailed in priority, importance, and length of execution. You may choose to use automated testing tools; in most cases, this automated approach should be augmented with human testing as well.

5.6.3 Deciding to Go Live

Ultimately, the decision to start running the business on the new solution must be made internally, and taken seriously. As you approach the milestone of a new OIM solution, make sure that the team has enough information to enable a defensible “go” or “no go” decision to be made. The go live checklist created earlier in the upgrade process should be leveraged to verify that the success criteria have been achieved during the project.

Make sure that all affected groups from both business and IT are represented in this decision. If you have a formal steering committee, this will be the appropriate decision-making body. If for some reason there is no steering Committee, call a meeting for this purpose, gathering input from the stakeholders ahead of time and fostering the communication that will allow for an informed and broadly supported decision.

Similarly, a plan should be in place to allow a “roll back” to prior systems if critical stage gates are not achieved.

6 Conclusion

Upgrading from Oracle Identity Manager 9.1 to Oracle Identity Manager OIM 11gR1 can provide organizations with broad benefits to the scalability, manageability, security, and performance. However, there are numerous considerations that companies must bear in mind as they plan and execute their upgrade. Among these are not only the means and the path by which they will affect their upgrade, but also features that have changed or will change in the future that will impact their business needs.

Oracle provides tools to help companies successfully upgrade to Oracle Identity Manager 11g. Effective planning, testing, and proper use of tools can help make the upgrade a smooth experience that provides new capabilities to your organization.



Oracle Identity Manager: Migrating to 11gR1
March 2011

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 © 2010, 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.

0109