

Oracle Applications Release 12 Upgrade Project: An Overview of the Joys and the Pains

Marcella Sackett, Ph.D.
Traust Consulting

Introduction

Oracle Applications Release 12 was released in January 2007 with a completely redesigned Financials module and the addition of new features to the Manufacturing modules. The R12 applications are considered “The Global Release” for the E-Business Suite by implementing tools and functions that increase a corporation’s visibility to their organization from a global perspective. This is achieved through the introduction and/or enhancement of a Global Architecture and Global Business Functions.

This white paper is divided into two main sections: the first section provides a general overview of the new design, features, and functionality of Oracle Applications Release 12 (R12); the second section provides the insights and experiences of an upgrade project from 11.0.3 to R12.

Overview of Release 12: “The Global Release”

Release 12 of Oracle Applications presents entirely new methodologies, forms, processes, and functions to enhance the global view of a business’ financial and operational information. This global approach is obtained through two areas: Global Architecture and Global Business Functions. The Global Architecture is achieved through modifications in the areas of application functions and database structures. The Global Business Functions include thousands of new capabilities and the introduction of new modules that restructure and/or streamline business processes.

Global Architecture – Application Functions

Examples of the Global Architecture include the introduction of new application functions such as Multi-Organization Access Control (MOAC), Subledger Accounting (SLA), and Ledger Sets. It also includes database features such as additions to the Trading Community Architecture (TCA) and the Oracle Applications Tablespace Model (OATM).

Multi-Organization Access Control (MOAC) is a new feature in the E-Business Suite that includes the capability for a centralized access point to application functions for all operating units within an organization. Previously, if an end user needed to access data for different operating units, they needed to exit their existing screens and either use the change organization function or to change their responsibilities to a custom responsibility that pointed to a single operating unit. These options lead to either increased risk at data entry due to lack of visibility as to which organization is active or increased management and overhead of maintaining custom responsibilities for each operating unit. It also limits the end user’s capabilities to analyze information for all operating units. Thus, single-point access to all operating units through MOAC improves efficiency and global visibility.

Modules impacted by MOAC are varied yet emphasize where multi-operating unit visibility is beneficial, namely Payables, Procurement, Supply Chain Management, Order Management, Customer Data Management, and Receivables. The functions impacted within these modules are detailed in Table 1.

Area/Module	MOAC Functions
Payables/Payments	Funds Disbursements and Receipts
Procurement	Requisitions, Demand Purchase Orders
Supply Chain Management	Receiving
Order Management	Order Management
Customer Data Management	Trading Community Architecture (TCA) Credit Card Numbers
Receivables	Billing and Collections

Table 1. Modules impacted by Multi-Organization Access Control

An example where MOAC is now utilized is within the Payables/Payments module. When MOAC is enabled for a Payables/Payment responsibility, the data entry and the funds disbursement can be performed for all operating units in a centralized responsibility. However, MOAC is an option and is not required. This provides the system administrator with the flexibility of choosing to provide multi- or single-organization access through specific responsibilities. Another MOAC function available is the ability for a

receiver to receive product for multiple organizations; thus, further streamlining the warehouse processes.

The flexible security options provided by MOAC extend beyond the option of controlling access to the operating units at the responsibility level. It also includes new templates for a variety of functions within a responsibility to allow for single- or multi-operating unit access. An example is in the Payments responsibility where the Payments Administrator can choose to have payment batches created for one or more specified operating units. The combination of multiple operating units into a single payment batch increases efficiency and the flexibility of the templates provides flexible security.

Other areas of the application utilizing MOAC include reporting and several of the user interfaces (forms). Many reports now offer the option for single- or multi-operating unit execution by the addition of an operating unit parameter. Also, several user interfaces have also been updated to include an operating unit field so that the end user is well aware of within which operating unit s/he is working.

Subledger Accounting is another new feature of the Oracle Applications Release 12 and is the basis for the primary changes to the Financials modules. Subledger Accounting (SLA) provides a centralized repository for accounting transactions. All transactions pass through SLA and thus accounting models and rules are applied to each transaction in a consistent manner. In fact, now that the General Ledger data all passes through SLA, the accounting close process is much more streamlined across all of the modules. The close process now can happen via a single action for most modules. The benefits resulting from the addition of the SLA changes (see Table 2) include increased auditability and the ability to be compliant with accounting rules from both the local and the corporate perspectives. The increased auditability is derived from the fact that SLA contains complete drill down capabilities from the General Ledger to individual transactions. For instance, the end user can start at a summarized, Payables journal entry and drill down via SLA all the way to the individual Payables invoice transactions.

Benefits of Subledger Accounting (SLA)	
Benefit	Achieved Via
Corporate Rules Compliance	Accounting Standardization
Local Rules Compliance	Improved Local Compliance
Increased Control	Common Data Model Data Model and Repository
Increased Auditability	Full Drill Down Capabilities
Streamlined Closing Process	Common Posting Engine
Global Compliance	Multiple Accounting Representation

Table 2. Benefits of Subledger Accounting

Another benefit driven by the consistency of the data in the SLA is the ability to report the data through multiple representations. These representations are derived through Ledgers and Ledger Sets (see Figure 1). These new reporting capabilities replace the traditional “Set of Books”. A Ledger is touted as the “repository of financial truth” in which all of the transactions contain the same 4 c’s: accounting method, chart of accounts, calendar, and currency. Ledger Sets, on the other hand, are a grouping of Ledgers that must have at least the same chart of accounts and calendar. These Ledger Sets provide the corporation with the ability to view their global financial information in a single, cohesive unit. Through the use of Ledgers and Ledger Sets of data in SLA, an organization is readily able to comply with corporate accounting rules for standardization, local accounting rules for improved compliance, and improved global compliance.

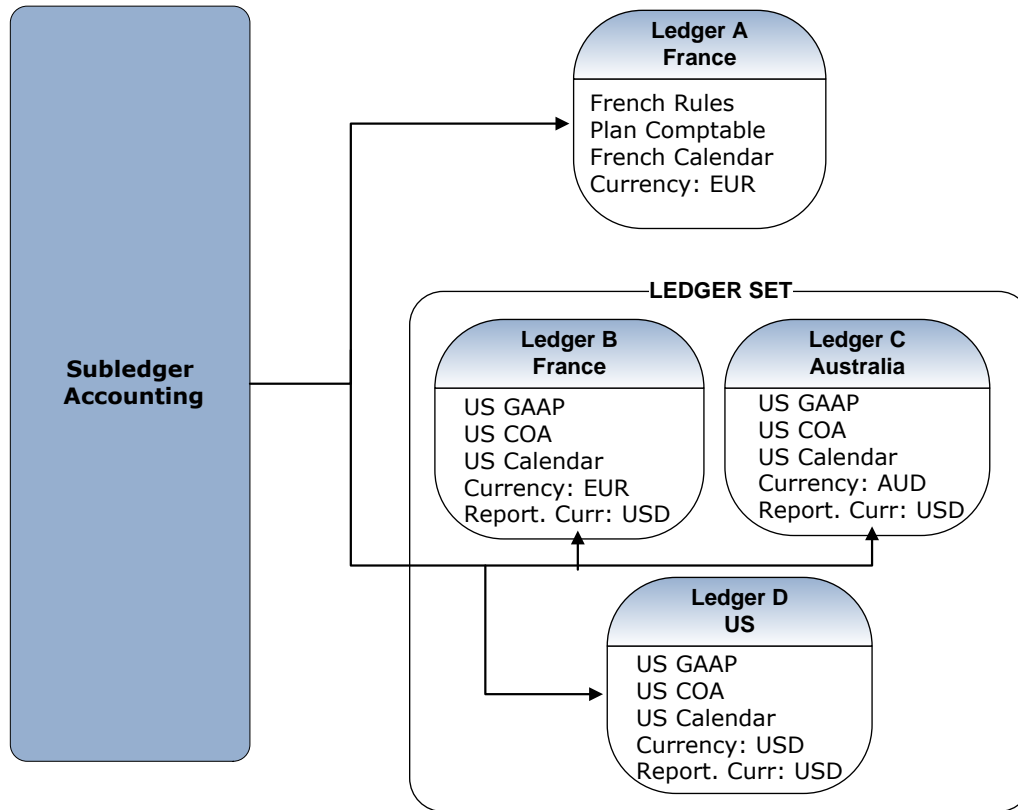


Figure 1. Ledgers and Ledger Sets

Global Architecture – Database Structures

The global architecture, as it pertains to database structures, was further expanded in R12 from the basis established in 11*i*. The Trading Community Architecture (TCA) now encompasses the Payables module by including supplier data and the Cash Management module as it pertains to both internal and supplier banks and bank accounts. In fact, it should be noted that the banking setups have been moved to the broader, new module of Payments. Thus, the TCA now includes the following “parties”: customers, suppliers, employees, and banks. Inclusion of these parties within the TCA facilitates the maintenance of information for the suppliers and the banks across all operating units. One change can be automatically propagated to all operating units. In order to facilitate this movement towards TCA, Oracle has included views of the original tables to assist with any backwards compatibility concerns.

The Oracle Application Tablespace Model (OATM) was first introduced as an option with 11i applications and continues to be encouraged as part of the R12 applications. This model consolidates the 300+ tablespaces within the database down to a total of 12 tablespaces with one tablespace for the base tables, one tablespace for index management, and 10 tablespaces for database management such as the undo, temporary, and system tablespaces. The reduction in the number of tablespaces contains several benefits including an increased ease in management and efficient space utilization.

Global Business Functions

In conjunction with the enhanced Global Architecture associated with Release 12, there are also several new enhancements to the business functions within the applications. These new features include several new modules, the release of new tax and bank models, and enhancements in the manufacturing modules. Some of the new modules include: Payments, E-Business Tax, Sourcing Optimization, Project Portfolio Planning, Service Desk, Case Management, Clinical Data Repository, Customer Loyalty, Utility Biller, Financial Consolidation Hub, and Healthcare Intelligence. In addition to new modules, Oracle Release 12 contains over 2,400 additional new capabilities.

With Release 12 being a global release, Oracle has also redefined some of its models in order to accommodate this globalization. Two of the main areas of change include the new tax model as part of the E-Business Tax module and the new banking model. E-Business Tax is the centralization of all of the tax information within the E-Business Suite. This new model is so all-encompassing that it includes all of the modules that contain tax information except for Payroll tax. For instance, some of the modules that utilize the new tax engine are Purchasing, Internet Procurement, Payables, Inter-Company Invoicing, Order Management, Receivables, General Ledger, and Internet Expenses. This centralized repository of tax information increases the ease of maintenance, especially when it is combined with the third party options of either Vertex or Taxware. However, the third party additions are not required and tax can still be manually maintained and customized to meet the needs of each corporation. Another benefit of the new E-Business Tax module is that it contains a Tax Simulator function which allows the end user to test transactions containing tax without creating actual transactions within the system.

A new banking model is also introduced as part of Release 12. This new model ties bank accounts to the legal entity level rather than to each operating unit. In addition to increased ease of maintenance, the new model supports the new Payments module. In this module, a single payment instruction can be created for each bank now that the banks are set at the legal entity level rather than the operating unit level. Thus, a single payment instruction, previously termed a payment batch, can be created for the organization as a whole for all payments transacting through the same bank.

The Financials area is not the only set of modules to be impacted in the globalization of the E-Business Suite. Enhancements in the manufacturing areas include the introduction of the Manufacturing Execution System (MES); a converged, single source of global inventory for mixed-mode manufacturers; a third party option of Demantra; and Transportation Management. The Manufacturing Execution System (MES), designed for both discrete and process manufacturing, is the “one stop shop” for shop floor operations. It includes a centralized location for WIP, Quality, Time and Labor Tracking, on-line instructions and requirements for individual operations, and a clock in/clock out capability.

The converged, single source global inventory model for mixed-mode (discrete and process) manufacturing now stores inventory information in a common data model to provide a global overview for inventory. Of the two modes of manufacturing, the process flow mode is the most impacted. Also, this globalization still requires the creation of separate organizations for the process and the discrete data.

Two additional areas of opportunity include the option to use the Transportation Management module or the point solution of Demantra. Transportation Management assists with freight cost reduction and the benefits of this module include that it is 100% web-based, is highly configurable, and can support a global business. Demantra, which is not part of the E-Business Suite, is an advanced demand management tool. This tool assists with the integration of the often hard to manage areas of demand planning such as trade promotion management and “real-time” sales operations planning.

Summary of the “Global Release”

With the release of Oracle Applications Release 12, the E-Business Suite (EBS) is now truly a “Global Release”. This lofty goal was achieved primarily through a complete redesign of the Financials modules

and the addition to or enhancement of a variety of functions. These changes result in an increased efficiency in daily tasks, new functionality, new terminology, new processes, and the ability to have a global view of the corporation.

Upgrade to Release 12: The Joys and Pains

Summary of the Upgrade Project

The E-Business Suite upgrade project, used as a case study for this paper, was performed for a mid-sized, discrete manufacturing firm in Minneapolis. The upgrade was significant going from Oracle Applications 11.0.3, RDBMS 8.0.6 to Oracle Applications Release 12, RDBMS 10g. The footprint of the existing Oracle Applications at the client was relatively small and consisted of General Ledger, Payables, Inventory, Bill of Materials, Material Planning/MRP, Engineering, Purchasing, and Cash Management. The number of customizations in the system was minimal; however, there existed one critical customization that utilized the Demand Interface. This interface fed information from a custom, legacy order entry and shipping system into Oracle Applications to generate demand and to relieve inventory.

Project Overview

Decision to Upgrade to Release 12

Prior to choosing to upgrade to Release 12, the client considered upgrading to 11.5.10. Several factors were taken into consideration when R12 became an option. For instance, the stability of 11.5.10 was weighed against the opportunities available in R12. The decision to move to R12 rather than 11.5.10 was based primarily by considering the long-term, strategic vision of the company which includes significant growth and expansion. Another key aspect was that as an early adopter of R12 (project start date of May 2007), the client, who is relatively small-sized in the world of Oracle Application users, would have the opportunity to utilize the Critical Accounts Support team within Oracle. The goal was to only use this team if deemed necessary due to slippage in timelines driven by issues in the application.

The primary benefit of upgrading to R12 was that it fit with the strategic vision of the client. Prior to deciding to upgrade the Oracle Applications, the client had spent time re-evaluating Oracle as the ERP software solution. They also spent time designing the future state of their IT infrastructure to match and

accommodate the strategic vision of the corporation as a whole. After making the selection to stay with Oracle Applications for their ERP system, the client designed a multi-year rollout to migrate their legacy system components over to the E-Business Suite. Since the client was starting with a relatively small footprint of modules that did not yet include some of the larger, more complex modules such as Order Management and Accounts Receivable, it was decided that upgrading the smaller footprint would be more straightforward than upgrading these larger modules at a later date. Each of these individual factors led to the decision to upgrade to Release 12.

Module Footprint, Infrastructure Plans, and Team Structure

The footprint at this client was relatively small with 8 modules in 11.0.3 including General Ledger, Payables, Inventory, Bill of Materials, Material Planning/MRP, Engineering, Purchasing, and Cash Management. Due to the changes in the structure between the 11.0.3 and the R12 application, four additional modules were added to the footprint Payments, E-Business Tax, Legal Entity Configurator, and Subledger Accounting.

As part of the upgrade project, the client also integrated the purchase of new hardware that included a platform change from HP-UX to Linux. Another objective was to move from a 32-bit platform to a 64-bit platform.

The team to support the upgrade consisted of members from both the client and the consulting firm. The client team included a Project Sponsor, two IT staff members, and Subject Matter Experts (SMEs) in the areas of Finance, Purchasing, Material Control, Engineering, and Planning. The consulting firm provided the DBA, Developer, a Manufacturing Business Analyst, a Financials Business Analyst, and a Project Manager who also performed Business Analyst tasks. The focus of the consulting team was to create the new environment(s) as well as perform map and gap analyses of the functions, business processes, and reports on the existing version of the applications as compared to Release 12. Test cases and new business processes were also designed and documented. The reasoning for the consulting team to perform all of these tasks rather than the SMEs was to reduce the overall time commitment required by the firm's internal staff.

Original Project Timeline

The original project plan (Figure 2) set the timeframe of the project to be 6.5 months. The project consisted of five distinct phases: Analysis, Design, Solution Pilot, Construction, and Transition. The Analysis Phase included business requirements analysis, infrastructure planning, and the initial DBA planning for the upgrade. During the Design Phase, test cases were to be created and the initial DBA “playbook” for the upgrade completed in conjunction with a first build. The Solution Pilot was slated for the DBA to complete the first upgrade dry run (UDR) for the consulting team to use for detailed analysis of the R12 environment. Another deliverable during the Solution Pilot phase was that the consulting and the client teams were to begin the business process design for the cutover/go- live period. These plans included processes affected by the upgrade process and any other impacts to the business such as a communication plan. The Construction Phase, from September through October, included two additional upgrade dry runs (UDR-2 and UDR-3) along with User Acceptance Testing (UAT). The final period, the Transition Phase was slated for finalization of processes for the cutover by including a final practice simulation, actual “Go- Live”, and a two week period for stabilization.

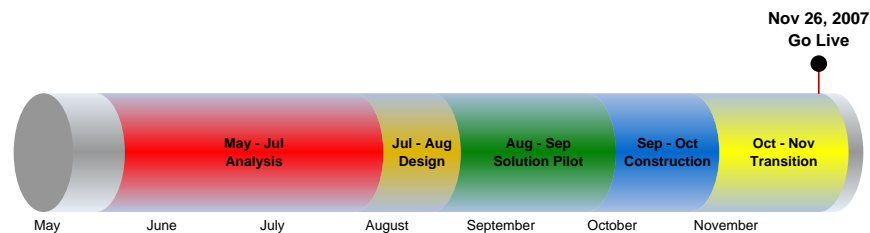


Figure 2. Original Project Timeline and Phases

The Reality of the Upgrade: The Joys and Pains

Timeline Struggles

The joys and pains that were to be experienced became evident early in the project. The first sign of pain was experienced through timeline slippage. The reasons for the early slippage were due to three main

issues: departure of the only DBA, migration to new hardware, and the difficulty of the first stage of the upgrade from 11.0.3 to 11.5.10.

Infrastructure Concerns

The departure of a key staff member is not atypical for a project. In this case, the turnover was not the only reason for the initial time slippage. The issue was that the time required to create the DBA “upgrade playbook” was going to take longer than originally estimated. The creation of the new environment was on the critical path of the project timeline because without a system containing actual, client data, it was difficult for the business analysts and the developer to meet their deliverable timelines. The delay in the “upgrade playbook” design was driven by two main factors: the migration to new platforms, especially the migration to 64-bit for both the database and the application tiers, and the upgrade steps from 11.0.3 to 11.5.10 (11.5.10 is required as a base application to move to R12). Risk mitigation came into play immediately after these issues were realized. Unfortunately, the only way to mitigate the risk of the issues surrounding the upgrade passing through 11.5.10 was to increase the timeline to allow for issue resolution. The issue of the 64-bit servers for both the application and the database tiers was able to be mitigated by choosing to upgrade on 32-bit servers for this project and then move to 64-bit at a later date in conjunction with another project.

Module Concerns

After the original risks associated with infrastructure were addressed, the second issue area that became evident was that there were concerns with certain modules, especially the Invoice Workbench within the Payables module. The root cause of the issues in this module stemmed from the complete restructuring of all of the Financials modules through the introduction of Subledger Accounting (SLA) and the introduction of Payables invoice lines. Issues with the invoice workbench and its underlying functions were so extensive that it was several months (approximately 4 months) before we could enter an invoice from beginning to end. The length of time was driven by the fact that the issues seemed to reside in multiple layers of the code so as soon as one issue was resolved through Service Request work, then another series of issues would arise at the next level. The concerns surrounding the Payables module were so extensive that we were able to leverage one of our original decision points for choosing to adopt

Release 12, namely Oracle's Critical Support team. We started working with a Customer Service Representative on a bi-weekly basis. The Customer Service Representative worked with a list of prioritized Service Requests to focus efforts within Oracle Development. After a period of time, it became evident to both the client/consultant team and the Oracle representative that the project required the attention of an additional layer deeper within the Oracle organization. It was at that point that the project was assigned to a Critical Accounts Manager who was able to leverage both Oracle's analysts and the development team. Through the efforts of the Critical Accounts Manager, Service Requests associated with the upgrade project were prioritized, managed, and escalated internally within the Oracle team. During this period of issue resolution within the Payments module, the Invoice Workbench form version level increased from x.x.23 to x.x.112. This level of version increase is indicative of the fact of the layers of complexity and thus the layers of issues surrounding the Invoice Workbench in the Payables module. However, these issues also led to the silver lining of being assigned a Critical Accounts Manager and learning and understanding in more detail the internal workings of Oracle.

Final Timeline

The timeline slippages due to the issues that arose were out of the control of the project team. Adding additional resources to the project would not have sped up the resolution of the issues involving the Payable module. The project timeline became reliant upon the Oracle development team to resolve issues as quickly as possible.

The new timeline (see Figure 3) included two extensions totaling 3 months along with an additional upgrade dry run to the timeline. These extra phases resulted in the addition of 3 months to the original schedule for a total of 9.5 months. The benefits of extending the timeline were extensive: increased user time on the new system, increased practice time for the business processes during the cutover to the new system, increased number of practice rounds for the DBA, and resolution of nearly all open issues.

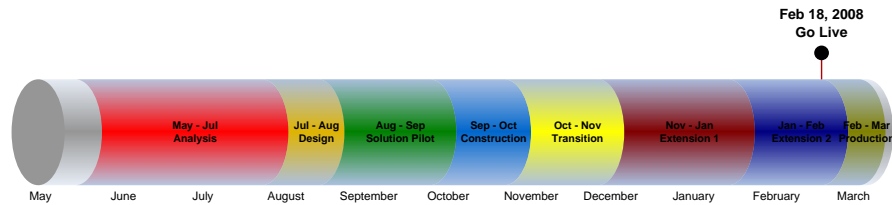


Figure 3. Final Project Timeline and Phases

Thoughts/Recommendations on R12 Upgrades

Having successfully completed an R12 upgrade, there are several key thoughts and recommendations to present to other Oracle customers who are considering upgrading to R12. The recommendations vary from preparatory steps to the planning of the project timeframe.

Project Timeframe Considerations

As with any project, there are several criteria and points to consider when determining a project timeline. These include the size of the current footprint as compared to the R12 footprint, customizations/extensions in the system and where they reside in the application, and the team members' experience level with R12.

Current Footprint vs. Future (R12) Footprint

In the case of upgrading to Release 12, one area of consideration is the current size of the module footprint as compared to the final size of the upgrade footprint. This project involved the relatively small initial footprint of 8 modules, but with the addition of new features and functionality in R12, a total of 12 modules were part of the upgrade. The additional modules that were implemented (E-Business Tax, Subledger Accounting, Legal Entity Configurator, and Payments) could be part of almost any R12 project. These new module setups and configurations must be learned. The primary learning methods utilized on this project were implementation manuals, user guides, and frequent Online Web Conferences (OWC) with analysts from Oracle (see "Experience with R12" section for more details).

Customizations

In addition to overall footprint size, other key areas for consideration are customizations and extensions. In this project's case, customizations to the system were focused on an interface between a legacy order entry system and the Demand Interface to build demand in Oracle Applications. Due to the obsolescence of the Demand Interface as of 11i, additional design, development, and setup of the backend of Order Management were needed to feed information into MRP.

Customized reports and forms need to be evaluated to determine how they are impacted by the redesign of the application. For instance, a custom report on supplier information is unlikely to require modifications because Oracle has created backwards compatibility to the TCA through views with old table names (ie. po_vendors is now a view that feeds and reads information into the TCA for suppliers). However, if a customization was created on the suppliers form, then development and/or gap analysis time needs to be added to the project plan. The reason is that suppliers are now entered and maintained through a web-based form rather than the traditional forms structure. In fact, many areas impacted by new functionality have been migrated to web-based forms, and while these forms provide a variety of available personalizations, they may not fulfill the entirety of the business need.

Other areas to be considered when creating a project timeline are data conversions or custom data feeds into the Financials modules. These modules and the underlying data structures have been significantly updated with the addition of the Subledger Accounting Module and should be reviewed in detail.

Experience with R12

The significant changes to R12 means that even Oracle Applications experts with extensive knowledge of the Oracle E-Business Suite are required to re-learn and learn anew many of the standard features within the applications. For instance, navigation paths and setup paths have changed. This may seem like a small and expected change as a part of any upgrade; however, the changes and new features in R12 are extensive. Lack of familiarity can lead to an increase in time for the creation of test cases because new navigation paths and business processes needed to be learned and documented by the analysts.

New module setups are also a factor in determining the project timeline. Documentation manuals associated with these setups were in their infancy at the time of this project (as of late 2007), and therefore, were not extensive enough to complete setups without the creation of a Service Request. Service Requests often led to the need and/or request for an online web conference (OWC) with the analyst. The first OWC would often lead to a second OWC in order to have the lead analyst on the phone to assist the frontline analyst. The need for a second OWC became less and less frequent over time as the frontline analysts became more familiar with R12. The infancy of the manuals and the duplication of effort of the OWCs led to an increased timeframe for setups of the new modules.

Preparatory Steps

Fortunately there are preparatory steps that organizations can take in order to reduce the project timeline and to increase the overall smoothness of the upgrade to Release 12, specifically Functional Team preparation, the Technical Team preparation, and the end user training.

Functional Team Prep

The basis of Functional Team preparation is to read and practice. The reading needs to occur at multiple levels starting with the Implementation Guide which contains specific information for the Subledger Accounting and the method for upgrading your data. Next to be read is the Upgrade Guide. Finally, the individual module guides, as usual, are a must read.

The second area of preparation is to practice, practice, practice. Team members need to learn the new modules and the new navigation paths within the areas that have been updated. The setup of a Vision environment pre-project followed by the setup of a R12 environment with live data will increase the navigation comfort level and knowledge base of the Functional Team.

Technical Team Prep

The Technical Team also has areas of preparation, including new technology training (Forms 10, Fusion Middleware, Application Developer Framework, BI Publisher, and JDeveloper).

Forms Customization

There are still plenty of traditional forms based on Forms 10 within the E-Business Suite. However, it is obvious with this upgrade that Oracle is migrating most of the newly created/revised forms to self-service forms. Self-service forms utilize the OA Framework and thus have a wide range of possibilities for personalizations such as the ability to add fields, hide fields, and to make fields required. Prior to attempting to customize these forms, it is important that the OA Framework personalization options be reviewed and analyzed because it is much more difficult to customize a self-service form versus the traditional forms. Self-service forms are more difficult to customize because they include java classes, XML files, “metadata files”, and JSP files.

Reports Customization

Standard reports in R12 are based on Reports 10g. However, in a fashion similar to the forms, Oracle started the migration of new reports and some previously existing reports into BI Publisher (previous XML Publisher) reports that are based on XML or XSL-FO. A large number of rtf templates are provided for reports based in BI Publisher. Though most templates are easily customized, some still require some XSL-FO coding. The benefit of utilizing these templates is that there are several output options including PDF, Excel, HTML, Word, and PowerPoint, and as long as fonts are installed on the servers, then there are several font options.

Summary of Lessons Learned

The main areas and aspects in the realm of “lessons learned” include the following: the upgrade, the number of version releases between the current application version and the upgrade; reports in the Financial modules; training; and the setups for Financials.

The Upgrade

The first lesson learned is in regards to the upgrade itself, specifically the number of version releases increased from current level to R12 level. This project started with the baseline of 11.0.3, contained an interim step at 11.5.10, followed by the upgrade to R12. It is the first of these steps from 11.0.3 to 11.5.10 that was the longest and the most difficult portion of the upgrade. The goal was to make this part

of the upgrade consistent and streamlined yet that was not possible. Each upgrade dry run generated new issues in this step that needed to be problem-solved and analyzed. On the other hand, the upgrade from 11.5.10 to R12 was extremely streamlined with minimal issues. An option that should be considered is the Maintenance Wizard, an Oracle tool that assists with upgrades and patches. The Maintenance Wizard was not selected for this project since it requires an additional database installation and was not desired by the client.

When upgrading from 11.0.3 (or earlier) to R12 with an interim step at 11.5.10, it is imperative that a working version of the 11.5.10 applications be left intact in order to problem-solve upgrade issues. During this project, issues were discovered in the SLA and elsewhere that utilize the new accounting event structures such as GL Drilldown, AP and PO Reconciliation report, and the Accounts Payable Trial Balance. It would have been much easier and more time effective to have a working interim 11.5.10 environment. Once we had an 11.5.10 environment that correlated with our working R12 environment, the Oracle Corporation analysts and developers were able to assist with a gap analysis of the data and create one-off patches to resolve the issues.

Reports

The R12 release is primarily a re-write of the Financials modules, and thus it can be expected that most of the Financials reports are also affected. Many Financial reports were migrated to BI Publisher format since the code needed to be reworked due to the new data structures. Also, many reports and concurrent requests have new processes in place to reflect the addition of Subledger Accounting. Unfortunately, the result of these changes is that there were many financial-related reports that initially did not work including the AP and PO Reconciliation report, the Purchase and the Invoice Price Variance reports, the Accounts Payable Trial Balance, and the 1099 report. These reports are required for monthly reconciliation and the client selected them for validation of the upgraded data. Consequently, these reports became part of the critical path for the success of the project. Reports are rarely in the critical path and are often an after thought in end user testing even when the project team encourages thorough testing of reports. Even though these reports have been fixed by Oracle, it should be noted that testing time should be increased to insure that all reports are thoroughly tested and are working as expected.

Training

It is recommended training time be increased over a typical upgrade project. This is especially applicable for the Financial end-users since these modules are so greatly affected by the upgrade to R12. Users should have hands-on activity prior to User Acceptance Testing (UAT) so that they are comfortable with the new processes and navigations. An area that is typically difficult for end users is the toggling between traditional forms and self-service forms. For example, Subledger Accounting provides the opportunity for complete drilldown capabilities from a summarized journal entry all the way down to an individual transaction line. This drill down involves switching from traditional form to self-service and back to tradition forms (see Figure 4).

The figure consists of five screenshots from an Oracle financial system, illustrating the navigation between traditional and web-based forms. Red arrows and numbers 1 through 5 point to specific features in each screenshot.

- 1:** A traditional form titled "Find Journals" with fields for Batch, Journal, Ledger, Source (Payables), Category (Purchase Invoices), Period (JAN-08), and Control Total. A red arrow points to the "Find" button.
- 2:** A traditional form titled "Enter Journals (MATE PRECISION TOOLING)" showing a table with columns for Batch Status, Source, Category, and Period. A red arrow points to the "Review Journal" button.
- 3:** A traditional form titled "Journals (MATE PRECISION TOOLING)" showing journal entry information and a table of lines. A red arrow points to the "Line Drilldown..." button.
- 4:** A web-based form titled "View Journal Entry Lines" showing journal entry details and a table of subledger journal entries. A red arrow points to the "View Transaction" button.
- 5:** A traditional form titled "Invoice Workbench (General Ledger Super User)" showing a table of invoice transactions. A red arrow points to the "View Payments" button.

Figure 4. Navigation between traditional and web-based forms. 1. Journal post find screen (traditional form); **2.** Journal summary screen (traditional form); **3.** Journal entry information with drilldown capabilities (traditional form); **4.** Journal Entry details (web-based form) with transaction drill-down capabilities; **5.** Individual transaction entry (traditional form).

Summary

Oracle Applications Release 12 is truly a “Global Release”. New features and designs such as Subledger Accounting, Ledger Sets, and Multi-Organization Access Control provide option is for a global perspective of an organization with streamlined, efficient methodologies. Nonetheless, as with any release that includes substantial redesigns and modifications, there come issues and the need for new approaches and business processes. From an upgrade project perspective, new approaches include training for end users, functional analysts, and the technical team. In addition, increased project time should be taken into consideration for training of end users; validation and testing of reports; and the increase in size of the module footprint that is driven by the R12 restructuring. Overall, the Release 12 application is a powerful, newly designed application that continues to improve the efficiency and options for global organizations.