

ORACLE®

**Running Oracle's Global Single Instance (GSI) on
Oracle E-business Suite 12. *Lessons Learned in
Upgrading to Financials Release12***

Nayyar Siddiqui
Senior Director, Applications IT

R12 Upgrade: the approach



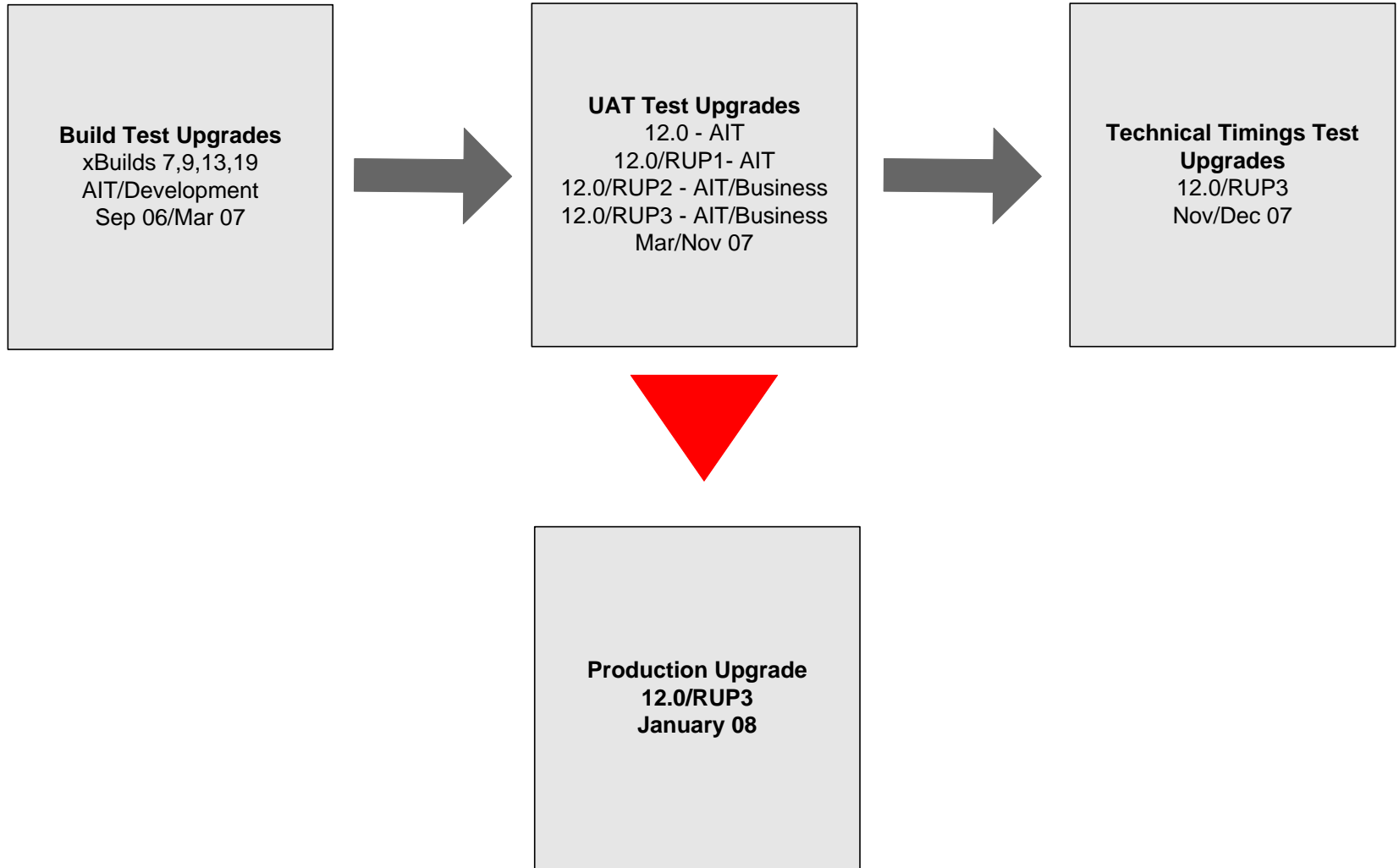
R12 Upgrade: Approach

- Simplify by approaching the project as a ‘technical upgrade’
- Only implement new functionality if it is mandatory – Ebtax, SLA, Ledgers, Payments module
- Migrated only 8 years of SLA data versus all 20 years
- Validate test plans to ensure up to date and include all flows and scenarios
- Test your scenarios with transactions created in R12 as well as those upgraded from 11i e.g. test AR credit memo creation for a 11i invoice as well as an R12 invoice
- Take the opportunity to review and standardize set-ups and responsibilities and eliminate unnecessary customs and reports prior to upgrade in order to reduce effort to upgrade and the variations to test
- Ensure load/volume testing is carried out

R12 Upgrade: Approach (contd.)

- Identify features that could be enabled prior to R12
 - Advanced Collections replaces the 11i collections workbench. Global roll out prior to the R12 upgrade
- Identify features that must be used at go live
 - TCA for Suppliers and Banks
- Identify features that could be a go live feature but not mandatory if issues arise during UAT
 - Multi-Org Access Control (MOAC) for Purchasing Responsibilities
 - MOAC allows users to process, access and report on data for multiple Operating Units within a single responsibility
- Identify features that must be Post-R12 due to complexity
 - Ledger Sets (you need a standard Chart of Accounts to do this)
- Identify all custom code/features that can be replaced with standard product
 - e.g. procurement approvals (custom) replaced by AME
- Copy of the system intended to be read only was made available to the business for reference purposes only

GSI R12 Timeline



Key product areas to focus on



Lessons Learned

- At Commencement of Upgrade:
 - Clear workflow transactions pending approvals; minimize the number of 'in-transit' transactions prior to starting the upgrade
 - Clear interfaces and any process exceptions as far as possible
 - Delete pending concurrent processes prior to the upgrade in case resubmitted inadvertently after upgrade
 - Update request sets for R12 changes
 - Review custom incompatibilities after the upgrade to ensure upgraded correctly
 - Generate key reports to enable pre and post upgrade data verification and reconciliations

Lessons Learned (contd.)

- Sub-ledger Accounting (SLA)
 - SLA is the 'interface' that sits between the sub-ledgers and GL and is a 'behind the scenes' module, largely transparent to end users
 - All sub-ledger accounting data is stored in SLA and is imported to GL from SLA
 - There must be a protocol in place to ensure all the sub-ledgers are transferring their data into SLA in a consistent manner
 - When reconciling your sub-ledgers to GL, important to note that all the reconciliation reports point to the SLA tables

Lessons Learned (contd.)

- Sub-ledger Accounting (SLA) (contd.)
 - Standard reconciliation reports work off the SLA tables (however please note if you have custom reconciliation reports these will need to be 'reworked' to work off the SLA tables)
 - The upgrade will leave AR and PA auto-accounting rules intact and the migration will create default SLA rules
 - The sub-ledgers store the 'default' accounting, SLA holds the 'true' accounting. SLA is the 'source of truth'
 - For example, Invalid Code Combination ID's (CCID's) can now be corrected using the 'alternate account code form', however these 'alternate' CCID's are reflected in SLA, not in the sub-ledger tables

Lessons Learned (contd.)

- GL and invalid code combinations
 - R12 provides a feature for updating invalid code combinations. It is the 'Alternate Account code form'
 - At the time of importing to GL, invalid CCID's are flagged and the data is not transferred to GL. Use the form to fix the invalid CCID
 - Also important that the segment inheritance program is run after all segment changes. This ensures that data once imported to GL cannot be posted to GL (scenario where a segment change has been made after the data is imported to GL)
 - *Note: the update only happens in SLA so the source data is not corrected. SLA is the source of truth*

Lessons Learned (contd.)

- Create Accounting
 - This is the program that is used to transfer data to GL. Data is transferred to SLA and imported to GL
 - These programs 'kick off' child programs. AR for example kicks off 5 child programs
 - Ensure your environment is configured to handle multiple requests to prevent locking situations where the programs do not complete
 - Run create accounting for sub-ledgers right through the month so as not to deal with large volumes at month-end
 - If you dealing with large volumes, please note that you can run this program by category e.g. by receipts or by invoices
 - Consider parallelization of the create accounting program to improve performance

Lessons Learned (contd.)

- Create Accounting (contd.)
 - Review the output of the sub-ledger period close exceptions report. This provides a clear view of the exceptions
 - Understand all the status codes that are on this report and the actions to be performed to address each code

Lessons Learned (contd.)

- Global Accounting Engine (AX)
 - This is replaced by SLA, which is now the standard for all feeders (AR, AP, FA, PA) even for non AX countries
 - All statutory Sets Of Books now become secondary ledgers with automatic posting between the primary ledger and the secondary ledger
 - For chronological sequencing countries, need to setup the Doc#sequence, Account#sequence, reporting#sequence. The reporting sequence is assigned to both the sub-ledger and general ledger journal entries when the GL period is closed
 - Ensure all statutory postings are up to date prior to starting the upgrade
 - Reports are all XML based. Need XML 'know how' to debug issues

Lessons Learned (contd.)

- Ebtax
 - Tax setup is migrated from 11i to R12
 - We created scenarios in R12 (simple, medium, complex) to prototype (and help learn & understand ebtax)
 - For accounts payables ensure you test your invoice sources (XML, isupplier, flat file, manual). Each of these should automatically derive the regime to rate information (mandatory R12 attributes)
 - Clean up your customer and supplier tax setup before the upgrade. This will help significantly to ensure tax is derived correctly
 - We use VERTEX for US tax calculations and the LAD tax engine for Brazil, Argentina and Colombia

Lessons Learned (contd.)

- Payments
 - All payments format files are now XML
 - The 11i files will be obsolete
 - GSI uses ABN, citibank and Wells Fargo. We have had to create XML payment files for all our banking partners
 - Do not underestimate the time required for this work
 - Successful testing will requires sign-off from the banks
 - All ipayment setup is migrated to the payments module

Lessons Learned (contd.)

- TCA
 - There is a new form for entry of customer data that looks and feels very different than the 11i Customer Standard form
 - Suppliers and banks are now also stored in the TCA data model
 - Backward compatible views no longer exist for AR; change any critical customs
 - Location & address validation is now done via the Geography Name Referencing (GNR) process, this replaces the old AR-based location data model and requires some detailed understanding to configure optimally
 - Run the GNR process to map addresses in location tables to 'master reference geographies'; Ebtax uses this mapping information when calculating tax

Lessons Learned (contd.)

- iProcurement and iexpense
 - Cleanup all catalog data; this is a pre-upgrade step but allow time for this
 - Cleanup supplier data to reduce the number of parties created in TCA; PD provide scripts to help identify those that can be cleaned up
 - We implemented MOAC for those responsibilities that are new with R12 and are 'meant' to be MOAC e.g. supplier profile administrator for creating/managing suppliers
 - 11i standard per diem and mileage schedules are migrated to R12. However detailed testing early on is recommended
 - Any expenses or requisitions that are not manager approved at the time of the upgrade will error in R12. The user will need to withdraw and then resubmit

Lessons Learned (contd.)

- Inter-company
 - Clean up the existing inter-co rules; the cleaner the rules, the easier to complete the setup
 - Internally we use many transaction types to derive intercompany accounting, each with different initiator and receiver rules
 - The transaction types are migrated to R12, however need to setup the distribution and clearing accounting for auto-approved and non auto-approved transactions
 - For example, the recipient cost center is derived from the transaction type and the company code is derived from the recipient Legal Entity company code
- Customs
 - Know your list and validate if they are important to the business
 - With R12, many reports are now in XML, so the extract is provided with a default layout which can easily be modified by the customer
 - ramp up XML knowledge
 - mod PL/Sql technical stack not supported with R12

Other considerations



Other considerations

- Month End Protocol
 - Ensure there is a published timetable that communicates the month close schedule and the 'workdays' on which each process is to be completed
 - Run create accounting daily for all your sub-ledgers; do not let this 'stack up' till the end of the month
 - Ensure a request set is created to include create accounting and the 'transfer to GL'
 - Review the output of the sub-ledger period close exceptions report. This provides a clear view of the exceptions
 - Understand all the status codes that are on this report and the actions to be performed to address each code

Other considerations (contd.)

- Ebtax and SLA are new 'shared' modules between feeders; establish a protocol for sharing issues, bugs and bug releases with all the concerned sub-ledgers
- While the upgrade takes care of the migration to R12, depending on how many migrations & upgrades your 'instance' has been through, you will find data issues
- Project Accounting: Post upgrade run the 'UPG: Upgrade transaction attributes' program for historical data so that existing transactions in PA can be adjusted (converted 5 years of data)
- For AP, remember the payments module and invoice workbench are significantly reworked, so do not underestimate end user training
- Scrambling of credit card data in test instances can effect the run of a few migration script/programs provided by dev for credit card data migration from OM to Oracle payment
- Volume test critical processes to find performance issues!

Critical success factors



Critical success factors

- Buy-in from all stakeholders
 - Executive Management Support and Sponsorship
 - Business Partnership
 - Product Development focus and Involvement
- Detailed project planning and communications
 - Multiple rounds of upgrades
 - Plan Project Management – involve right teams, reporting of issues, regular status reports
 - Organization of testing and report issues
 - Need detailed inventory of patches and time taken to apply

Critical success factors (contd.)

- Strong user involvement in all phases of the project
 - Business process changes due to eBTax, SLA etc to be defined upfront
 - Clarify business responsibilities
 - Adequate Business training – at Oracle, Global Process Owners from the business were responsible for this
- Focus on Testing
 - Multiple rounds of testing. Focus on new/changed areas
 - Comprehensive and detailed test plans (tax scenarios, run create accounting in parallel)
 - Robust End to end testing during UAT with close and reconciliation activities carried out
 - FINALLY, ensure all reconciling items are documented prior to the upgrade. Ensures a 'clean' starting point