This article appeared in the winter 2013 issue of *OAUG Insight* magazine, the official publication of the Oracle Applications Users Group, and is reprinted with permission.

Working with Custom Code in E-Business Suite Upgrades

By Mark Clark, Senior Partner, O2Works

Dealing with custom code is one of the biggest challenges of moving to a newer release of Oracle E-Business Suite (EBS). When planning the overall upgrade effort, an important variable of project timelines and budgets is understanding the amount and type of customizations associated with the production footprint.

The upgrade from EBS 11i to EBS R12 is a serious project, requiring a dedicated team and a well-constructed strategy that includes generous testing and issue-resolution phases that should be fully staffed, supported by the executive leadership and not compromised. Planning for and completing the analysis work on customizations can prevent headaches during the training, testing and issue-resolution phases of the upgrade.

In the strictest sense, customization involves altering standard EBS functions. Custom code can be expensive to develop but even more expensive to maintain. For this article, however, I want to discuss custom code and related objects not delivered or supported by Oracle, such as:

- Reports, interfaces, extensions, customizations and workflows.
- Alerts, form personalizations, folders and the custom library (custom pll).
- Third-party bolt-ons and packaged software that augments EBS.

Minimize EBS custom code

Try to eliminate as much custom code as possible. Research the more recent releases to see if new functionality replaces the homegrown code.

Determine what custom code is being used and phase out custom code that is no longer required. Set up queries or monitor the concurrent requests to see if reports and even interfaces are being used. If possible, this should be done over several months to get the most insight.

One customer moving from EBS 11.5.10 to R12 was able to retire 20 percent of its custom reports. Some of the custom reports supported a phased rollout and were no longer needed when the rollout was completed. Some were written when the users needed extra help and insight as new users; once they had been live for a number of years, these users had found other ways to meet business requirements. Others were made obsolete by data marts that were implemented after the go-live.

Migrating obsolete custom code to work with EBS R12 is a waste of scarce resources. Global companies may see that the improved localization functionality in R12 can allow them to eliminate some custom code as well.

Migrate only what's necessary

If you have taken copies of standard Oracle reports and modified them, the best practice is to take a new copy of that object from the new release and reintroduce the modifications. Don't spend inordinate amounts of time trying to get an old 11i report to work in an R12 environment.

The same recommendation holds true for workflow items. You can save a great deal of time and effort by taking a clean, plain-vanilla version from the latest version of the software and re-applying some of the tweaks and changes that were made. This should also save time during issue resolution in the testing cycles.

If your user community has folder functionality set up on forms, odds are very good that these will need to be rebuilt. Best practice involves deleting all folders across all modules. Folders can be recreated in R12 and can actually be migrated between instances to save labor. If you don't organize your folders on all your EBS modules, some of the issues that users encounter in initial testing have to do with form folders that don't make sense in the newer version of the software because of new or changed forms, or even changes in the underlying table structure.

" Try to eliminate as much custom code as possible. Research the more recent releases to see if new functionality replaces the homegrown code. "

While many standard open interfaces and APIs are typically not heavily changed during the migration to EBS R12, you may rename interfaces and add new or additional parameters. While the main interface may not have been greatly modified, you should test additional lookups or validations or extra data transformations that were added for your specific site, as other supporting tables may have been changed.

You should retest event and periodic alerts, as underlying tables may have changed during the upgrade. Event alerts are typically disabled during the upgrade, so you may need to re-enable them prior to testing.

Forms involve tricky custom code

Customizations to Oracle Forms can be tricky, as they are probably one of the most invasive customizations besides database triggers. If your business processes require modifications to the forms, replace them with personalizations as much as possible. If you do have customized forms, look to see if those forms will exist in the new, plain-vanilla version. Some EBS modules have quite a few brand-new forms.

You should be able to replace some old forms with new OA Framework-based forms. If you still require customizations, enlist an expert to help. Many issues can be introduced by non-industry standard changes to OA Framework forms.

On a related subject, I discourage trying to leverage old custom menus, as there is a likelihood that those menus point to obsolete forms and functions. By starting with a freshly delivered menu from Oracle and making changes, you can help reduce issues that arise during testing phases.

Finally, consider the third-party products that you may have previously licensed. You may be spending extra money for functionality that has now been introduced into the applications. Examples include third-party form and label software, check printing software, invoice printing and delivery solutions, and more.

" Customizations to Oracle Forms can be tricky, as they are probably one of the most invasive customizations besides database triggers. "

Oracle EBS customers that thoroughly consider their homegrown custom code and do the advance work on customizations during the analysis phases of an upgrade can save considerable time, money and aggravation during their migration.

Mark Clark, a senior partner with O2Works LLC, has extensive experience working with customers to plan and execute their upgrades to Oracle E-Business Suite Release 12. Mark is also a member of the OAUG Board of Directors, and served as the OAUG president from 2011 through 2012.

SearchOracle

This article was originally published by SearchOracle.com, a TechTarget publication, and is reprinted with permission.