



Development - Standards

Oracle Development – Part 1: Are Your Customizations Protected?

By Cheetah Solutions

Editor's Note: *In this first of a series of three white papers, the author presents an overview of creating a custom application and naming custom objects; both are critical steps to protecting your Oracle® customizations from patch applications, upgrades, and migrations. In future white papers, coding standards and guidelines will be discussed.*

Overview

To preserve your Oracle Applications module customizations across different upgrades, migrations, and patch applications, and for ease of maintenance and administration, a custom application with appropriate data object nomenclature should be implemented for the Oracle Applications. Oracle Corporation fully supports the registering of custom applications, and the procedure developed is based on Oracle's recommendation.

All customizations are loaded in a separate directory on the server, thus separating standard objects from custom objects, and all custom application objects are registered under one custom application.

Custom objects typically include: applications, concurrent programs, forms, request groups, responsibilities, and menus. Database-specific custom objects include tables, views, and scripts that feed concurrent programs and control data transfers from other applications in and out of the Oracle database interface tables.

Benefits

Why register your customizations to a custom application?

- Future Oracle Applications upgrades are facilitated if the volume of custom enhancements is kept to a minimum and are well documented.
- Future custom enhancements can be more easily developed and launched into production sooner if standards are available as guidelines for developers to follow and adhere to.
- Support and maintenance of custom enhancements can be minimized, reducing headcount.
- Mission critical custom enhancements are protected against accidental over-write during routine patch applications and release upgrades.
- The process results in documentation of all customizations.
- Redundant code is reduced.

Things to Consider

Custom code is cumbersome and risks being over-written during a patch application or a release upgrade because:

- Non-standard naming conventions exist.
- Custom objects are named the same as Oracle standards.
- Custom code is not uniquely identified – a future release of Oracle could adopt the same name for standard code.

This could result in costly re-engineering of code and down production time.



Development - Standards

Oracle Development – Part 1: Are Your Customizations Protected?

.....More Benefits

A custom application reduces the risk of custom code “accidentally” being over-written during patch applications and upgrade projects which would result in a recovery from storage (if available) and/or a re-engineering of the custom code.

- It reduces the amount of maintenance performed by System Administrators, freeing them to address critical technology and business issues instead.
- It reduces the uncertainty for developers creating the code, resulting in fewer cycles to launch a customization into production.
- It reduces the effort end users must make to remember the name of and query on custom objects, allowing them to become more productive.

Data Locations

Registration of multiple applications requires considerable effort on the part of support personnel to create, support, and maintain.

Multiple locations for custom code buried either in layers of UNIX sub-directories or under PVCS project names, for example, are very cumbersome and create difficulties identifying what is what.

Custom code not located properly runs the risks of being over-written during patch applications and release upgrades and/or being lost or unavailable during a disaster recovery event, all of which can result in down production time and costly re-engineering of code.

Custom Application

A custom application requires the contributions from Oracle DBAs, System Administrators, and Operating System Administrators for each instance of the Oracle Applications. At a minimum, the steps required are:

1. Create an environment variable in the operating system for the custom application to point to.
2. Define sub-directories for the basepath.

Note: The sub-directories for the custom \$TOP should mimic the sub-directories for the standard Oracle Applications \$TOPs, assuming the operating system layout for the Oracle Applications follows the standards for Oracle Flexible Architecture (OFA).

3. Register the custom application in the Oracle Applications.

Prerequisites:

- Define an environment variable that translates to your application’s basepath.
- Set up a directory structure for your application.
- If your application resides in a database other than the database where Oracle Alert resides, create a database link.

4. Register the Oracle ID for the custom application.
5. Create User in SQL*Plus and assign tablespace and grants.



Development - Standards

Oracle Development – Part 1: Are Your Customizations Protected?

6. Assign the custom application to a data group.
7. Registered all objects in the Oracle Applications for the custom application.
8. Place execution files for the objects in the custom basepath.

.....And Still More Benefits

Because custom applications and basepaths are high maintenance for support personnel, it is recommended that the number of custom applications and basepaths be kept to a minimum. All custom objects can be registered under one custom application that points to one custom basepath, and the directory structure for the custom basepath can mimic the Oracle standard.

- It reduces the amount of effort required by System Administrators to register new custom code freeing them to address critical technology and business issues instead.
- It reduces the amount of maintenance performed by System Administrators, freeing them to address critical technology and business issues instead.
- It removes multiple layers of sub-directories, reducing the risk of “losing” custom code.
- It is in line with Oracle’s recommended Oracle Flexible Architecture (OFA), which will facilitate the quality of support received from Oracle Support Services.

Redundancy and Documentation

Oracle customers, often times, have several similar sounding custom reports that are nothing more than multiple versions of the same reports and/or objects that are essentially the same with a different sort option or parameter.

The key fear is that this potential redundancy may contribute to a slower month end and fiscal closing if these reports are all queued to the Concurrent Manager, which in turn affects the performance of the Oracle environment and the productivity of the end users.

..... It Just Keeps Getting Better Benefits

Evaluate all customizations to determine the purpose for each and to determine if one or more can be grouped into a single customization. For example, reports can have multiple sort and parameter options at run-time rather than writing an individual report with hard-coded values for each specific requirement.

- It reduces daily data entry activities for end users because time is not spent switching back and forth between multiple forms, keying data that may be keyed into one master form.
- It reduces the fiscal close cycle for end users because time is not spent waiting for reports to finish printing so analysis can begin.
- It reduces the load on the concurrent manager because less processing is occurring which in turn improves the over-all performance of the Oracle environment.
- It facilitates future Release Upgrades because there is less custom code to consider for business processing re-engineering.
- It facilitates the GAP analysis efforts that will be required as part of future Release Upgrades.



Development - Standards

Oracle Development – Part 1: Are Your Customizations Protected?

Recommended Nomenclature for Custom Objects

Custom Forms

Oracle's nomenclature is two-fold:

1. The first three characters of an Oracle standard form identify the module it belongs to.
2. The last five characters identify the form.

e.g. APXINWKB.fmb
 | | |

 APX = Oracle Payables Module
 INWKB = Unique form name

Recommendations

- Name custom forms using either a three or four character prefix that is meaningful.
- Name custom forms using upper case, exclusive of filename extensions.
- Name custom forms using eight characters in length, exclusive of filename extensions.

If a customized form is a modification of an existing Oracle standard form, replace the first three letters of the form's name with the custom prefix. This will serve two purposes: it identifies the form as a customization, and identifies the Oracle standard that it has customized.

e.g. Oracle Standard Customization
 APXINWKB XXXINWKB
 | | |

 XXX = Custom Application
 INWKB = Unique form name

Custom Executables

- Name custom executables the same as custom execution files for custom concurrent programs.
- Name custom executable using upper case.

Syntax: Application Module Descriptive Suffix

e.g. XXXAPQRY
 | | | |

 XXX = Custom Application
 AP = Oracle Module
 QRY = Descriptive Suffix

Custom Execution File Names

Oracle's nomenclature is two-fold:

1. The first three letters of an Oracle Standard File Name identify the module it belongs to.
2. The last 5 letters identify the file.



Development - Standards

Oracle Development – Part 1: Are Your Customizations Protected?

e.g. GLXMDVDV.rdf
| | |

GLX = Oracle General Ledger Module
MDVDB = Unique file name

Recommendations

- Name custom execution files using a three or four character prefix that is meaningful.
- Name custom execution files using upper case, exclusive of filename extensions.
- Name custom execution files using eight characters in length, exclusive of filename extensions.

If a customized file is a modification of an existing Oracle standard file, replace the first three letters of the form's name with the custom prefix. This will serve two purposes: it uniquely identifies the file as a customization, and identifies the Oracle standard it has customized.

e.g. Oracle Standard = Customization
APXINWKB = XXXINWKB

If a customized execution file name has been developed from scratch:

Syntax: Application Module Descriptive Suffix

e.g. XXXAPQRY
| | | |

XXX = Custom Application
AP = Oracle Module
QRY = Descriptive Suffix

Custom Concurrent Programs

- Name custom concurrent program using either a three or four character prefix that is meaningful.

Syntax: Application Module Descriptive Suffix

e.g. XXX Payment Batch Report
| | | |

XXX = Custom Application
Payment Batch Report = Program Name

Custom Request Groups

- Name custom request groups using either a three or four character prefix that is meaningful.
- Name custom request groups using mixed case.

Syntax: Application Module Descriptive Suffix



Development - Standards

Oracle Development – Part 1: Are Your Customizations Protected?

e.g. XXX GL Inquiry
 | | | ||

 XXX = Custom Application
 GL = Oracle Module
 Inquiry = Request Group

Custom Responsibilities

- Name custom responsibilities using mixed case.

Syntax: Application Module Descriptive Suffix

e.g. XXX AP Sr. Admin.
 | | | |

 XXX = Custom Application
 AP = Oracle Module
 Sr. Admin. = Responsibility

Conclusion

There's no getting around it and that is a good thing. If you are going to take the time to develop custom objects for your Oracle Applications environment, then it is in your best interest to protect them against future patch applications, migrations and upgrades.

The information on our website and in our publications is the copyrighted work of Klee Associates, Inc. and is owned by Klee Associates, Inc.

NO WARRANTY: This documentation is delivered as is, and Klee Associates, Inc. makes no warranty as to its accuracy or use. Any use of this documentation is at the risk of the user. Although we make every good faith effort to ensure accuracy, this document may include technical or other inaccuracies or typographical errors. Klee Associates, Inc. reserves the right to make changes without prior notice.

NO AFFILIATION: Klee Associates, Inc. and this publication are not affiliated with or endorsed by Oracle Corporation. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Klee Associates, Inc. is a member of the Oracle Partner Network.

All other company and product names used herein may be trademarks or registered trademarks of their respective owners.

This article was originally published by Klee Associates, Inc., publishers of JDEtips and SAPtips.

For training, consulting, and articles on JD Edwards or SAP, please visit our websites: www.JDEtips.com & www.SAPtips.com.