

Package Versions in Oracle Applications

Author: Andy Rivenes, Lawrence Livermore National Laboratories, arivenes@llnl.gov

Introduction

This article will address how to resolve errors due to incompatible package versions in Oracle Applications. Most of the time the correct version is on your system and its just a matter of finding the right SQL script and running it against the APPS account.

Incompatible package versions occur because Oracle is not always able to define all prerequisites required or fails to perform the proper checks when installing new dependent code. This may occur when updated reports, forms or programs are added, or when installing or upgrading, and dependencies between products haven't been fully verified.

All installations at one time or another will come across this problem. Even character mode sites will see this, especially when new or patched reports are added. Typically if the character mode site has not loaded the SmartClient/NCA server side updates then when an incompatibility error does occur the odds of finding a script with a compatible package version are significantly reduced.

It is worth noting that Oracle provides most patches expecting that the latest versions of product maintenance have been applied.

Identifying Errors Due To Package Versions

Unfortunately this is typically the most difficult part of identifying a package version problem. Once an error occurs the following can be an indication of a package version problem:

- Missing function or procedure in a compiled package
- Invalid package which won't compile due to an invalid package dependency.

If a package cannot be compiled successfully it may be due to many factors, but if it can be determined that the failure is due to the code of the package (e.g. a missing procedure or function, invalid number of arguments, undefined variable) then it is very likely that either the current package, or a dependent package, is out of date. It should be noted that this scenario also occurs for tables and views, but that discussion is beyond the scope of this article.

The following is an error that was encountered in a Receivables report and is a typical type of error after applying a patch:

```
MSG-02000: ORA-06550: line 1, column 21:
PLS-00302: component 'REFRESH' must be declared
ORA-06550: line 1, column 7:
PL/SQL: Statement ignored
REP-0450: Unhandled exception ORA-06550: line 1, column 21:
PLS-00302: component 'REFRESH' must be declared
ORA-06550: line 1, column 7:
PL/SQL: Statement ignored
which was raised in a statement starting at line 26 of C_PP_AUTO_RULEFORMULA
```

This error is hard to catch before a runtime failure occurs. This is because the procedure REFRESH is missing from the package ARP_AUTO_RULE in the database. The only way to determine this is to either call Oracle Support, or to search the \$AR_TOP directory for package definitions with a REFRESH procedure. In this case we find a SQL script called arplbarl.sql in the \$AR_TOP/patchsc/107/sql directory. If more than one script has this procedure definition then the latest version should be used (based on the \$Header line).

It should be noted that this is still probably worth a call to Oracle support, but at least now the urgency has been reduced and a lot more information can be given to the support analyst.

Oracle Package Versions

Oracle packages always have a "\$Header" line identifying the source script and the version number of the package.

Database Source

The following script will list the \$Header line for all package specifications and package bodies owned by the APPS account. Obviously only the affected packages need to be listed, but this can provide a handy reference when chasing down multiple problems.

```
SELECT name || ' ' || text
FROM dba_source
WHERE line = 2
AND type LIKE 'PACKAGE%'
AND owner = 'APPS'
ORDER BY name, type;
```

Output will be of the format:

```
ACCRUAL_WRITE_OFF_SV/* $Header: POXACCWS.pls 80.0 96/10/26 22:06:13 porting ship $*/
ACCRUAL_WRITE_OFF_SV/* $Header: POXACCWB.pls 80.0 96/10/26 22:06:12 porting ship $*/
AD_DD/* $Header: addds.pls 23.6 97/01/17 12:16:47 porting ship $ */
AD_DD/* $Header: adddb.pls 23.7 97/01/17 12:16:46 porting ship $ */
```

Product File Source

At the operating system level the Oracle Applications product files can be searched for the product affected (e.g. \$AR_TOP in the previous example). The source files generally end in either .sql or .pls and can reside in any one of several SQL directories (e.g. ./install/sql, ./sql, ./upgrade/sql, ./patchsc/107/sql). Typically the beginning of the package definition will be of the format:

```
CREATE OR REPLACE PACKAGE ARP_AUTO_RULE IS
/* $Header: arplsarl.sql 70.10 97/10/20 16:42:12 porting ship $ */
```

On UNIX the utility *grep* can be used to find files that contain a specified string. Example to find all scripts that contain the string "refresh":

```
/applmgr/r10/ar/7.0.152/patchsc/107/sql$ grep -i refresh *.sql
arplbarl.sql: | refresh
<other output deleted>
arplbarl.sql:PROCEDURE REFRESH (Errbuf OUT VARCHAR2,
<other output deleted>
arplsarl.sql:procedure refresh (Errbuf OUT VARCHAR2,
<other output deleted>
```

The result indicates that the SQL scripts arplsarl.sql/arplbarl.sql contain the procedure "refresh". A find command can be used in UNIX to determine if there are any other scripts with the same name in the \$AR_TOP directory. This needs to be done to determine that latest version of the package on the system.

For the case where the script name is known (e.g. based on the package header line obtained from the source stored in the database) and just the latest version is needed the following can be used to loop through all of the possible directories (assumes korn shell):

```
cd $AR_TOP
for sqldir in `find . -type d -name sql`
do
  grep -i "<insert name here>" $sqldir/*.pls
  grep -I "<insert name here>" $sqldir/*.sql
done
```

The result will indicate all files with the string "insert name here" in them. Since Oracle places the version just after the file name on the \$Header line the result will also display all versions of the package. This can then be compared to what version is loaded in the database (see Database Source above). If the versions are different then it is very likely that the latest version will fix the problem. It should be noted that other dependent packages may still have to be recompiled.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.