PL/SQL Starter Framework

Release Notes

# Release 3.0 (February 2014)

The 2.0 release was mainly intended for 10g, but compiled and worked on 11g as well.

This 3.0 release was updated to take advantage of 12c features. It will not compile on 11g. Use version 2.0 for 9i through 11g.

## Scripts

* Updated scripts to ensure the install works on 12c. Must be connected to pdb, not root container when installing.
* Added README\_BEFORE\_INSTALL.txt file, which some folks intuitively expect.
* In case the script is being run on a linux/unix platform, fixed install script so it does not default to C:\Temp as the log path for the install log.
* Removed the sample TKT schema and added the missing sample BLG (Billing) schema. Remember that once you read the User Guide and understand how the app, app\_db, and app\_env tables work together, you may remove the TKT\_DEV, TKT\_TEST and BLG schemas and associated sample data in the framework tables.
* Commented out the Oracle Text install code which would have created a nice, searchable index on the content of the app\_email table. Everything works except for the actual index creation statement. There seems to be a bug which may be inherent in Oracle 12.1. This requires further investigation.

## Packages

* Altered a handful of packages to make use of the 11g feature to assign a sequence.NEXTVAL into variables, instead of selecting them from DUAL.
* Added drop\_sched and drop\_job routines to DDL\_UTILS
* Improvements to DDL\_UTILS.reset\_seq were copied to UTILS.reset\_seq.
* Added make\_clob\_list to STR
* Added the Tom Kyte functions to NUM: to\_base, to\_dec, to\_oct, to\_hex, to\_bin and my own bin\_to\_dec
* Added get\_db\_alias to LOGS.
* Removed the public specification and private implementation of ENV.get\_routine\_nm and ENV.line\_num\_here. Now favoring the new routines that use 12c features, like get\_my\_nm and get\_my\_line.
* Renamed who\_called\_me to get\_caller\_nm. Renamed who\_am\_i to get\_my\_nm. Re-factored to use UTL\_CALL\_STACK.
* Added get\_caller\_line and get\_my\_line to ENV.
* Commented out the private bundle\_stack\_lines and public caller\_meta from ENV. No more need for that parsing nonsense since UTL\_CALL\_STACK was released.
* Modified API\_APP\_LOG to fill the new call\_stack and error\_stack columns. If the database is 12c or higher, it will use the slightly more specific 12c call stack (derived from the new UTL\_CALL\_STACK package). Also removed the line\_num parameter.
* Modified IO so it is not hard-coding the package or proc names for logging. Instead, the routines are identifying themselves using ENV.get\_my\_nm which is newly based upon the 12c UTL\_CALL\_STACK features.

## Triggers

* Altered the audit triggers to make use of the 11g feature to grab the next chg\_log\_id directly from app\_chg\_log\_seq.NEXTVAL, instead of selecting it from DUAL.
* Removed the BEFORE INSERT triggers whose sole job was to ensure the PK column was filled before insert. This duty is now handled by the 12c sequence.NEXTVAL default added to the table’s PK columns.

## Tables

* Updated the surrogate key PK columns so they are consistently INTEGER instead of NUMBER. This ensures you can’t have PK values with decimal precision, just nice whole numbers.
* Updated the PK columns to use the new DEFAULT <sequence>.NEXTVAL ON NULL feature of Oracle 12c, and removed the now unnecessary AFTER INSERT triggers whose sole job was to fill the PK if not given.
* Updated all NOT NULL columns with a named constraint so that we avoid system-named unique constraints, thus saving us from false positives when comparing one install of the framework to another across databases.
* Added call\_stack and error\_stack to APP\_LOG to separate these calling contexts from developer-written log messages. Removed the line\_num column. The line number where logging call was made can be found in the call\_stack column. For messages with an attendant error stack, the line number where the error actually occurred can be found in the error\_stack column.

## Known Issues

Basically the MAIL package is broken. Not sure if it’s my installation of 12c, or a missing wallet, or what. Will require further investigation. Luckily the uptake on 12.1 hasn’t been speedy.

* Could not get MAIL.is\_smtp\_server\_avail to work. When called from older clients like PL/SQL Developer 10, I get
* Could not get MAIL.send\_mail with LOB attachment to work. One 11g I’m receiving

ORA-29532: Java call terminated by uncaught Java exception: java.lang.NullPointerException

ORA-06512: at "APP\_CORE.MAIL", line 133

ORA-06512: at "APP\_CORE.MAIL", line 597

ORA-06512: at line 22

On 12c I’m getting

Updating email 3 as Error due to SMTP Return Code [1] Msg:

javax.mail.SendFailedException: Sending failed;

nested exception is:

javax.mail.MessagingException: IOException while sending message;

nested

exception is:

javax.activation.UnsupportedDataTypeException: no object DCH

for MIME type multipart/mixed

# Release 2.0 (April 2010)

## Scripts

* Fixed install scripts to operate more smoothly, and work on 11g.

## Packages

### API\_APP\_LOG

* Minor bug in text being concatenated into the log text. Now wrapping line\_num with TO\_CHAR.
* Added COMMIT to the trim table API.

### TYP

* Expanded mime type subtype to 100 chars.

### UTIL

* str\_to\_bool: added support for converting NULL
* num\_to\_bool: improved handling of non-zero numbers
* Minor fixes to some mime types in get\_mime\_type and get\_otx\_doc\_type.

### EXCP

* Changed the signature of the assert() routine, removing need to pass in the calling routine name. Sorry.
* Revamped any framework packages that were relying on the old assert signature, removing their need for a local name variable in each as well, simplifying things.

### STR

* Added two new string-related constants: NUL and LFCR.
* Added the string centering function, named ctr()

### MAIL

* store\_mail and send\_mail: Fixed bug in i\_attach parameter in that the default needed to be NULL, not EMPTY\_CLOB().
* Store\_mail: fixed bug, now checks for length of the attachment, ensuring it is greater than zero.

### IO

* Fixed side effect of 9i bug in get\_file\_props().

### ENV

* Added vld\_path\_format, set\_app\_cd()
* Added more useful and shorter tag() and untag(). If module, action and info aren't passed, they'll be filled automatically with the PL/SQL unit/package, routine and line number respectively.

### LOGS

* Clarified set\_dbg() comments.
* Added the 10g backtrace functionality to the main msg routine.
* With the enhancement to the env package to get the caller's metadata, was able to add the ability of these logging routines to get their own caller, rather than needing it passed in all the time. So routine name and line num are now very much optional parameters.

### DT

* Added UNIX\_EPOCH constant, dt\_to\_epoch and epoch\_to\_dt functions.
* Rearranged the order of the parameters in get\_time\_diff\_str to match that of its complement, get\_time\_diff.

### DDL\_UTILS

* Corrected several bugs in reset\_seq()
* Fixed side effect of 9i bug in analyze\_table()
* Added MATERIALIZED\_VIEW as one of the object types visited by recompile()

### APP\_ENV

* Added app\_version column to track which application version is running on the associated environment.

## New Objects

* Added standalone drop\_all\_obj, meant to be compiled in schema with SYSDBA privs. Can drop all objects from a named schema.

## Sample Application(s)

* Added the reports1.sql and reports2.sql scripts which were used at IOUG 2010 and UTOUG TD 2010 to demonstrate the benefits of developing with a PL/SQL framework.

## Test Suite

* Updated to work with package changes. Note: the test suite works with PL/SQL Developer v7 and higher.