Administrative Tasks

Background

Definitions

In the context of Oracle Workspace Manager, to *merge* data in a workspace means to apply changes from a child workspace to its immediate parent in the workspace hierarchy. To *refresh* means the opposite: to apply changes from a parent to its child.

CED Workspace Naming Convention

CED workspaces whose name begins with an underscore character (e.g. _dev) are considered to be administrative workspaces and are not advertised via the API *CEDdb->stages()* function call or the *ced -workspaces* command line invocation. Otherwise, the same rules apply to them as to other workspaces.

Continually refreshed workspaces

A workspace that is continually refreshed will automatically be refreshed when data is committed to or merged into its parent workspace. A workspace may be defined as continually refreshed when it is created or at a later point in time, however once a workspace is designated continually refreshed, that feature may not be disabled without removing and recreating the workspace. It's also important to note that once a row has been modified in a child workspace, that row will no longer receive automatic updates from the parent workspace.

Based on experience, the implementation of continually refreshed workspaces by Oracle causes too much trouble when merging tables with non-primary key uniqueness constraints. As of June 2013, the child workspaces of _dev are no longer set to continually refreshed**.**

Some useful queries and commands related to continually refreshed workspaces are illustrated below.

```
/* Query to find out which workspaces are continually refreshed */
SQL> select workspace, continually_refreshed from all_workspaces;
WORKSPACE                      CONTINUALLY_REFRESHED
------------------------------ ---
LIVE                           NO
IOCDev                         YES
MagnetDev                      YES
STAGE                          YES
MyDev                          NO
_dev                           NO

/* Make a workspace continually refreshed if it is not already */
SQL> EXECUTE DBMS_WM.ChangeWorkspaceType('MyDev');

/* Specify continually refreshed at Workspace Creation Time */
SQL> EXECUTE DBMS_WM.CreateWorkspace(workspace=>'IOCDev', isRefreshed=>true);
```

Merging a workspace

The instructions in this section deal specifically with the workspaces in the CED3_DEVL schema.

Note that in our multiple-schema configuration, it is in fact, the CED3_OPS schema that we treat as the "live" or "production" version of the data. Therefore, **simply merging data into the LIVE workspace of CED3_DEVL does not make it visible to Operational tools**. Making data operational involves an additional step of copying the data to CED3_OPS schema which will be discussed later in this document. It is also important to make sure that the CED3_DEVL LIVE workspace has been updated with any LIVE-EDITS that may have been made in CED3_OPS.

Case 1 - merge STAGE into _dev

Follow these steps which will require access to a terminal on a Control System Linux workstation:

1. Run the ced_audit command to verify the consistency of the STAGE workspace
2. Execute DBMS_WM.mergeWorkspace procedure to merge STAGE into _dev
3. commit or rollback the merge
4. Run the ced_audit command to verify the consistency of the _dev workspace

```
#Audit the STAGE workspace
%>  ced_audit -wrkspc STAGE -f /dev/stdout -e -p

# Connect to sqlplus as ced3_devl
%> sqlplus ced3_owner@ceddb01

/* Now merge the pending changes from STAGE into _dev */
SQL> EXEC DBMS_WM.mergeWorkspace('STAGE', auto_commit=>false);

/* If there was no error, commit the changes, else rollback;
SQL> commit;

/* exit sqlplus */
SQL> exit

#Audit the _dev workspace
%>  ced_audit -wrkscp _dev -f /dev/stdout -e -p
```

Case 2 - merge _dev into LIVE

Follow these steps which will require access to a terminal on a Control System Linux workstation:

1. Run the ced_audit command to verify the consistency of _dev workspace
2. Compare the modify_date timestamps between CED3_DEVL LIVE and CED3_OPS LIVE.
3. If CED3_OPS has a more recent timestamp, it means there have been LIVE-EDITS that must be copied to CED3_DEVL. Run the CED_ADMIN.copyWorkspace procedure to pull those changes into CED3_DEVL LIVE.
4. Execute DBMS_WM.mergeWorkspace procedure to merge _dev into LIVE
5. commit or rollback the merge
6. Run the ced_audit command to verify the consistency of the LIVE workspace

```
#Audit the _dev workspace
%>  ced_audit -wrkspc _dev -f /dev/stdout -e -p

# Connect to sqlplus as ced3_devl
%> sqlplus ced3_devl@ceddb01

/* Copies the OPS Data into current workspace, changing only differing rows */
SQL> EXEC CED_ADMIN.copyWorkspace('LIVE','CED3_OPS');

/* Now merge the pending changes from _dev into LIVE */
SQL> EXEC DBMS_WM.mergeWorkspace('_dev', auto_commit=>false);

/* If there was no error, commit the changes, else rollback;
SQL> commit;

/* exit sqlplus */
SQL> exit

#Audit the LIVE workspace
%>  ced_audit -f /dev/stdout -e -p
```

Case 3 - Selectively merge a single element from IOCDev into _dev

There will be times when it is not desirable or possible to merge all of the changes in a child workspace into its parent. This is frequently the case right now with the IOCDev workspace where a developer may request that a single new IOC be made available to OPS, but it is not desirable to merge the entire IOCDev workspace which also contains multiple incomplete IOCs.

Case 3a - Using the web

The simplest way to merge a single IOC from IOCDev into _dev is to use the web-based merge tool provided as part of the CED web application (Figure 1). Note however, that the web-based merge only merges element property values and is not appropriate if changes have been made to property definitions in the catalog (cmpnt_type_*) tables. The web tool also has no capacity to deal with conflicts. If a conflict prevents the merge from completing successfully, it will be necessary to resort to manual intervention via sqlplus to resolve the conflict.

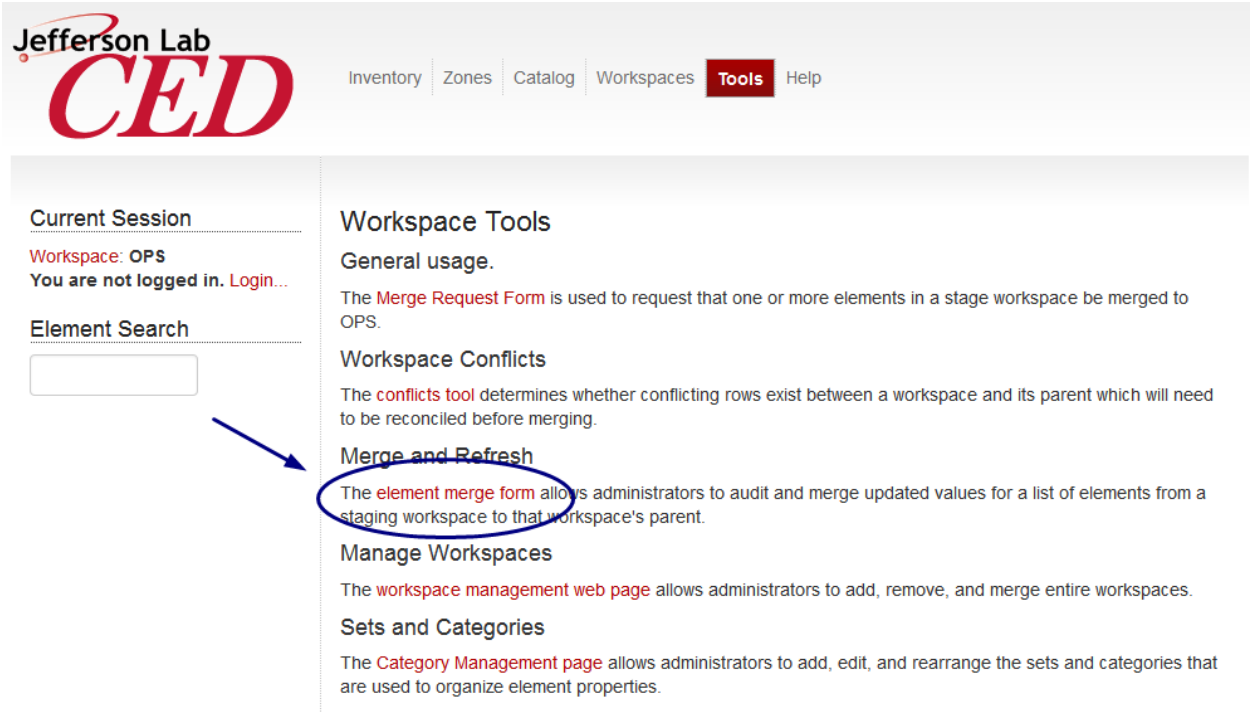

Figure 1 - Accessing the web based merge.

The merge tool requires authentication as a CED administrator and will prompt the user to log in if he/she has not already done so. The next step is to select the child workspace from which elements will be merged using the provided drop-down menu and then provide the list of element names in the text area. The element names may be separated by commas or new lines.

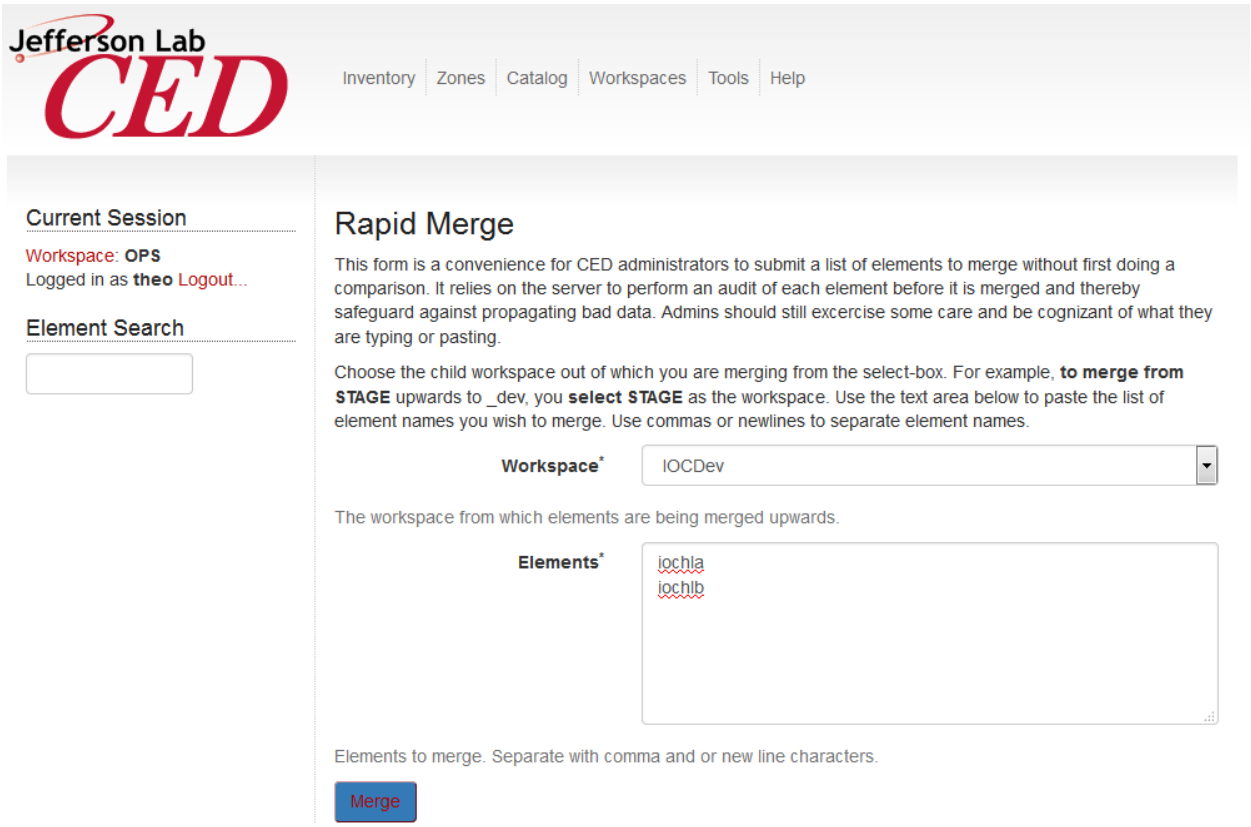

Figure 2

Case 3b - Using Sqlplus

```
#Audit the element(s) in the workspace
%>  ced_audit -wrkspc IOCDev -standard -scope iochla

# Connect to sqlplus as ced3_devl
%> sqlplus ced3_devl@ceddb01

/* Now merge the element(s) */
SQL> EXEC CED_ADMIN.mergeElementByName('IOCDev', 'iochla');

/* If there was no error, commit the changes, else rollback;
SQL> commit;

/* exit sqlplus */
SQL> exit

#Audit the _dev workspace
%>  ced_audit -wrkspc IOCDev -standard
```

Dealing with conflicts

The following types of conflicts can arise during a merge or refresh operation

- A Workspace Manager conflict can occur if the same row of data has been updated in both the parent and the child workspace.
- A CED conflict can occur if the same element or property has been created independently in two workspaces. The two elements or properties will have different primary keys, but share the same name. Attempting to merge or refresh the two workspaces together will result in an integrity constraint violation because of the duplicated name.

```
/* The following error message was encountered while merging elements via the web page */
EXCEPTION: Failed to merge iocse12BpmCrate from STAGE to _dev.

ORA-20055: conflicts detected for workspace: 'STAGE' in table: 'CED3_DEVL.CMPNT_PROP_FK_CMPNT'
ORA-06512: at "WMSYS.LT", line 7172
ORA-06512: at line 1


ced3_devl@CEDDB01> column WM_WORKSPACE format A12
ced3_devl@CEDDB01> select * from cmpnt_prop_fk_cmpnt_conf;

WM_WORKSPACE   CMPNT_ID      VALUE CMPNT_TYPE_PROP_ID   VALUE_ID       DIM1       DIM2 WM_DEL
------------ ---------- ---------- ------------------ ---------- ---------- ---------- -----
STAGE             14351      16129               2390                     0          0 NO
BASE              14351      14359               2390     313026          0          0 NO
_dev              14351      14359               2390     313026          0          0 YES

/* In this case the property was deleted in _dev and updated in STAGE.  We want to keep the
value from STAGE which is the CHILD of _dev */

SQL> EXECUTE DBMS_WM.BeginResolve ('STAGE');

PL/SQL procedure successfully completed.

SQL> EXECUTE DBMS_WM.ResolveConflicts ('STAGE', 'CMPNT_PROP_FK_CMPNT', 'CMPNT_ID=14351',
'CHILD');

PL/SQL procedure successfully completed.

SQL> COMMIT;

Commit complete.

SQL> EXECUTE DBMS_WM.CommitResolve ('STAGE');

/* Now we can return to the web page and try the merge again and not get the error. */
```

For more discussion of versioning and dealing with conflicts, see the Wiki topic at:
https://devweb/twiki/bin/view/AHLA/CEDVersioningImplementation

Copying data between schemas

Copying Data from CED3_DEVL to CED3_OPS

Once data has been merged into the CED3_DEVL.LIVE workspace, it will often be desired to place the data into CED3_OPS where it will become the new production "OPS" (AKA "LIVE") data.

```
# Connect to sqlplus as ced3_ops
%> sqlplus ced3_ops@ceddb01

/* Copies the OPS Data into current workspace, changing only differing rows */
SQL> EXEC CED_ADMIN.copyWorkspace('LIVE','CED3_DEVL');


/* exit sqlplus */
SQL> exit

# Audit the OPS workspace
%>  ced_audit -f /dev/stdout -e -p
```

Creating a Savepoint of CED3_OPS in CED3_HIST

Because CED3_OPS is not version-enabled for performance reasons, savepoints are instead stored in the CED3_HIST schema which is version-enabled. The process of creating a savepoint therefore entails copying the changes (if any) from CED3_OPS to CED3_HIST and then creating a savepoint in CED3_HIST.

The way to do this is by calling the CED_ADMIN.saveCEDOPStoHistory function. Because the user who invokes the procedure will be the owner of the savepoint, this function should only ever be invoked by the ced3_hist user. (The saveCEDOPStoHistory function has a check to make sure it's being invoked by the correct user, but why tempt fate?)

```
%> sqlplus ced3_hist@ceddb01

SQL> exec CED_ADMIN.saveCEDOPStoHistory;

11-NOV-2011 17:07: Beginning saveCEDOPStoHistory
11-NOV-2011 17:07: Beginning checkoutCEDOPS
11-NOV-2011 17:07: Beginning copyWorkspace (Local)
11-NOV-2011 17:07: create database link CEDSRC
11-NOV-2011 17:07: zones removed.... 0
11-NOV-2011 17:07: zones_links removed.... 0
11-NOV-2011 17:07: segments removed.... 0
11-NOV-2011 17:07: fk_cmpnt values removed.... 0
11-NOV-2011 17:07: cmpnt removed.... 0
11-NOV-2011 17:07: cmpnt removed.... 0
11-NOV-2011 17:07: cmpnt_type removed.... 0
(…)
11-NOV-2011 17:07: CMPNT_PROP_DATE added.... 0
11-NOV-2011 17:07: CMPNT_PROP_FK_CMPNT added.... 0
11-NOV-2011 17:07: CMPNT_PROP_FLOAT added.... 0
11-NOV-2011 17:07: CMPNT_PROP_INTEGER added.... 0
11-NOV-2011 17:07: CMPNT_PROP_STRING added.... 3
11-NOV-2011 17:07: cmpnt_prop_blob added.... 0
11-NOV-2011 17:07: cmpnt_type_owners added.... 0
11-NOV-2011 17:07: cmpnt_type_prop_owners added.... 0
(…)
11-NOV-2011 17:07: cmpnt_prop rows updated... 10
11-NOV-2011 17:07: cmpnt_prop_fk_cmpnt rows updated... 0
11-NOV-2011 17:07: cmpnt_prop_bool rows updated... 1
11-NOV-2011 17:07: cmpnt_prop_integer rows updated... 0
11-NOV-2011 17:07: cmpnt_prop_float rows updated... 1
11-NOV-2011 17:07: cmpnt_prop_string rows updated... 7
11-NOV-2011 17:07: cmpnt_prop_date rows updated... 0
11-NOV-2011 17:07: cmpnt_prop_blob rows updated... 0
11-NOV-2011 17:07: Beginning compareRowCounts ....
11-NOV-2011 17:07: MATCH: ZONES...SRC = 83 DEST = 83
11-NOV-2011 17:07: MATCH: SEGMENTS...SRC = 53 DEST = 53
11-NOV-2011 17:07: MATCH: ZONE_LINKS...SRC = 160 DEST = 160
11-NOV-2011 17:07: MATCH: CMPNT_PROP_FK_CMPNT...SRC = 8910 DEST = 8910
(…)
11-NOV-2011 17:07: MATCH: CMPNT_TYPE_PROP_DEF...SRC = 376 DEST = 376
11-NOV-2011 17:07: MATCH: CMPNT_TYPE_PROP_DIM...SRC = 338 DEST = 338
11-NOV-2011 17:07: MATCH: CMPNT_TYPE_PROP_REQ...SRC = 317 DEST = 317
11-NOV-2011 17:07: MATCH: CMPNT_TYPE_PROP_OWNERS...SRC = 0 DEST = 0
11-NOV-2011 17:07: End copyWorkspace
11-NOV-2011 17:07: Finished checkoutCEDOPS
11-NOV-2011 17:07: Create Savepoint AutoSP4 - Automatically generated savepoint
11-NOV-2011 17:07: Finished saveCEDOPStoHistory

PL/SQL procedure successfully completed.
```

Automated Nightly Savepoint

On dboracle which hosts the ceddb01 database, there is a crontab entry for the oracle user that executes the script */app/oracle/scripts/CEDhistsave.ksh* every night. This shell script calls the stored procedure CED_ADMIN.SaveCEDOPStoHistory to create a savepoint.

Restoring CED3_OPS from a savepoint in CED3_HIST

If it becomes necessary to replace the current data in OPS with data from an historical savepoint, this can be done using the CED_ADMIN package routine restoreCEDOPSFromHistory.

```
%> sqlplus ced3_ops@ceddb01

-- List out the savepoints if necessary

SQL> select savepoint, createtime, description from all_workspace_savepoints where owner='CED3_HIST' order by createtime;

 (…)
AutoSP77                  13-JUL-13
Automatically generated savepoint

AutoSP78                  05-AUG-13
Automatically generated savepoint

71 rows selected.

-- Let's presume we want to restore AutoSP77 from the July 13

SQL> exec CED_ADMIN.restoreCEDOPSFromHistory('AutoSP77');

PL/SQL procedure successfully completed.

-- Look at the admin_log table if you want to see more details about the operation:

SQL> select * from admin_log where logdate > sysdate -1 order by logdate;
```

Making Schema Changes

Oracle workspace manager places limits on the type of DDL that may be performed. See the Oracle workspace Manager documentation for details.
http://devweb/oradocs/appdev.111/b28396/long_intro.htm#i1009427

Exporting and Cloning

To CEDTEST for Development

It is often useful to make a copy of the production CED that can be used for testing, development, and debugging. The steps to do so are documented in the SysAdmin Wiki at https://devweb/twiki/bin/view/SysAdmin/HowToCloneProductionCEDDatabase.

Disaster Recovery

Starting up CEDDB01 on dbs

In the event of a sustained outage of dbo where the primary CEDDB01 database is hosted, the CED database can be brought online quicklin on the dbs server using the procedure found in the Sysadmin Wiki at:
https://devweb/twiki/bin/view/SysAdmin/HowToRestoreOracleOnStandbyServer

PHP Configuration

ini file

The following directives are placed in the file /usr/csite/pubtools/php/lib/php-web.ini

```
extension=cedlib.so
```

Symbolic link to cedlib.so

```
# Note that no-debug-non-zts-20060613 in the path below is dependent
# on the Zend Engine version of the PHP library.  It will vary between releases.

%> cd /usr/csite/pubtools/php/lib/php/extensions/no-debug-non-zts-20060613/

# Make a symbolic link to the desired PHP version of cedlib
%>ln -s /cs/certified/apps/cedlib/lib/rhel-6-ia32/php3.0/cedlib.so ./cedlib.so
```