Create Unit Tests

Unit tests should include the minimum number of tests to cover each scenario. For instance, for the following case statement

CASE src_cd = '24' THEN 'Japan' ELSE 'Others' END

the minimum number of tests is 2: for src_cd = '24' and for another.

Tests should be re-runnable and always return the same result.

Unit tests are stored on the github branch Unit-Test.

Naming Convention for Test File

A SQL script is created with following naming convention <Process_Type_Name>_<Group_ID>_<Process_ID>. (eg. PARTY_25_230_CTRL_T_PROCESS_TESTS).

This file will contain all the tests related to a particular Process_ID in a Group.

Creating a New Test

Identify the Process_Type_Name, Group_ID and Process_ID where the test need to be created

```
SELECT P.PROCESS_ID,P.ACTIVE,P.PROCESS_DESC,P.PROCESS_TYPE,P.PROCESS_TARGET

FROM ctrl_t_process_type t

JOIN ctrl_t_process_group g ON t.process_type_id = g.process_type_id

JOIN ctrl_t_process p ON g.GROUP_ID = p.GROUP_ID

where t.process_type_name = 'PARTY'

order by group_order, process_order desc;
```

If Test file is available for the Process to test upend the new test scenario to the file else create a test file.

Add Test file header information and Version information to specify which JIRA number is the test related to.

Please find a Sample Test File, PARTY_25_230_CTRL_T_PROCESS_TESTS.sql

```
-------------------------------------------------------------------------------------------------------
--
--  Title         PARTY_25_230_CTRL_T_PROCESS_TESTS.sql
--  Description   Tests for PARTY_25_230
--
--  Version     Date           Author          Description
--  -------     -----------    ------------    ---------------------------------------------------
--  1.0         02-May-2017    Jimmy Jose      Added Test-01 and Test-02 as part of Jira ADM_1234
-------------------------------------------------------------------------------------------------------


-------------------------------------------------------------------------------------------------------
-- Test ID : TEST-01
-- Test Description : 'First Name only Scenario';
-------------------------------------------------------------------------------------------------------
DELETE FROM CTRL_T_PROCESS_TESTS WHERE PROCESS_TYPE_NAME = 'PARTY' AND PROCESS_ID = 230 AND GROUP_ID = 25 AND TEST_ID =1 AND TEST_TYPE = 'U';

INSERT INTO CTRL_T_PROCESS_TESTS (PROCESS_TYPE_NAME, GROUP_ID, PROCESS_ID, TEST_ID, TEST_NAME, TEST_DESC, TEST_TYPE, ACTIVE, EXECUTION_DB, TEST_SQL, EXPECTED_VALUE)
VALUES (
    'PARTY' -- PROCESS_TYPE_NAME
    ,25     -- GROUP_ID
    ,230    -- PROCESS_ID
    ,1      -- TEST_ID
    ,'UT_ADM_T_DIM_PARTY_CLIENT_TYPE_INDIVIDUAL'    -- TEST_NAME
    ,'CLIENT_TYPE should be Individual when CLIENT_LAST_NM and CLIENT_ORG_NM ISNULL AND CLIENT_FIRST_NM IS_NOT_NULL '   --TEST_DESC
    ,'U'    -- TEST_TYPE
    ,'Y'    -- ACTIVE
    ,'ADM'  -- EXECUTION_DB
    ,'SELECT CLIENT_TYPE ACTUAL FROM ADM_T_DIM_PARTY WHERE PARTY_UK = ''GOALD#069001#9000024050'';' -- TEST_SQL (IT SHOULD HAVE ACTUAL AS ALIAS)
    ,'Individual'   -- EXPECTED_VALUE
);
```

Setup

Unit Tests can have Test setup, which is executed against specified Database before the Batch is executed for a Group or Process.

Please find a Sample Setup File, PARTY_25_230_CTRL_T_PROCESS_TESTSETUP.sql

```
--------------------------------------------------------------------------------------------------
--
--  Title         PARTY_25_230_CTRL_T_PROCESS_TESTSETUP.sql
--  Description   Test Setup for PARTY_25_230
--
--  Version     Date           Author          Description
--  -------     -----------    ------------    ---------------------------------------------------
--  1.0         02-May-2017    Jimmy Jose      Added Test-01 and Test-02 as part of ADM_1234
--------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------
-- Test ID : TEST-01
-- Test Description : 'First Name only Scenario';
--------------------------------------------------------------------------------------------------

INSERT INTO CTRL_T_PROCESS_TESTSETUP (TESTS_SETUP_ID,PROCESS_TYPE_NAME, GROUP_ID, PROCESS_ID, TEST_ID, ACTIVE, PROCESSSETUP, GROUPSETUP, EXECUTION_DB, SETUP_SQL, DESCRIPTION)
VALUES (
    NEXT VALUE FOR CTRL_T_PROCESS_TESTSETUP_SEQ_SETUP_ID    -- TESTS_SETUP_ID
    ,'PARTY' -- PROCESS_TYPE_NAME
    ,25      -- GROUP_ID
    ,230     -- PROCESS_ID
    ,1       -- TEST_ID
    ,'Y'     -- ACTIVE
    ,'N'     -- PROCESSSETUP
    ,'Y'     -- GROUPSETUP
    ,'LND'  -- EXECUTION_DB
    ,'INSERT INTO LND_T_FE_CLIENT(BATCH_ID,POL_OFFICE_CD,CLIENT_NO,CLIENT_TYPE_CD,REF_GROUP_CD,CONTACT_ID,ADDRESS_ID,CLIENT_STAT_CD,CLIENT_LAST_NM,LAST_NAME_EXTN,CLIENT_FIRST_
    ,'ADM_1234:UT_ADM_T_DIM_PARTY_CLIENT_TYPE_INDIVIDUAL'    -- DESCRIPTION
);
```

To create data setup, you can use CTRL_SP_GENERATE_INSERT procedure which will generate an insert statement and then you can change the data for your test purpose (more at the bottom of the page). This may be time consuming though if there are many tests or the table is wide. Then the quickest option is to use Excel to prepare data and write a formula to generate insert statements.

Data setup is used to inject single records which will generate expected test result. After the tests been executed, they should be deleted in Teardown (below) - make sure they have a unique identifier that can easily identify the injected records (for instance your story number).

Teardown

Unit Tests can have Test Teardown, which can be executed against specified Database after the Batch is executed for a Group or Process.

Please find a Sample Teardown File, PARTY_25_230_CTRL_T_PROCESS_TESTTEARDOWN.sql

```
--------------------------------------------------------------------------------------------------
--
--  Title         PARTY_25_230_CTRL_T_PROCESS_TESTTEARDOWN.sql
--
--  Version     Date           Author          Description
--  -------     -----------    ------------    ---------------------------------------------------
--  1.0         02-May-2017    Jimmy Jose      Added Test-01 and Test-02 as part of ADM_1234
--------------------------------------------------------------------------------------------------

--------------------------------------------------------------------------------------------------
-- Test ID : TEST-01
-- Test Description : 'First Name only Scenario';
--------------------------------------------------------------------------------------------------

INSERT INTO CTRL_T_PROCESS_TESTTEARDOWN (TESTS_TEARDOWN_ID, PROCESS_TYPE_NAME, GROUP_ID, PROCESS_ID, TEST_ID, ACTIVE, PROCESSTEARDOWN, GRO
VALUES (
    NEXT VALUE FOR CTRL_T_PROCESS_TESTTEARDOWN_SEQ_TEARDOWN_ID  -- TESTS_TEARDOWN_ID
    ,'PARTY' -- PROCESS_TYPE_NAME
    ,25      -- GROUP_ID
    ,230     -- PROCESS_ID
    ,1       -- TEST_ID
    ,'Y'     -- ACTIVE
    ,'N'     -- PROCESSTEARDOWN
    ,'Y'     -- GROUPTEARDOWN
    ,'LND'  -- EXECUTION_DB
    ,'DELETE FROM LND_T_FE_CLIENT WHERE POL_OFFICE_CD = ''069001'' AND CLIENT_NO = ''9000024050'';' -- TEARDOWN_SQL
    ,'ADM_1234:UT_ADM_T_DIM_PARTY_CLIENT_TYPE_INDIVIDUAL' -- DESCRIPTION
);
```

Whilst running your tests, you can disable teardown - it will then retain the data setup and makes investigating failures easier.

Executing Test

In order to execute Unit Tests as part of Batch API , provide –t “U”

nohup ./run_process_test.ksh -p "PARTY" -e "D" -t "U" –f "TRUE" > ../logs/party_run_`date +\%Y\%m\%d\%H\%M`.log 2>&1 &

In order to execute Acceptance Tests as part of Batch API , provide –t “A”

nohup ./run_process_test.ksh -p "PARTY" -e "D" -t “A” –f "TRUE" > ../logs/party_run_`date +\%Y\%m\%d\%H\%M`.log 2>&1 &

In order to execute a single test in a process without running Batch API

CALL CTRL_SP_RUN_TEST('PARTY',25,230,2,'UT_ADM_T_DIM_PARTY_CLIENT_TYPE_UNKNOWN','U')

In order to execute all the tests in a Group without running Batch API

CALL CTRL_SP_RUN_TESTGROUP('PARTY',25,'U')

In order to execute all the test in a process without running Batch API

CALL CTRL_SP_RUN_TESTPROCESS('PARTY',25,230,'U')

In order to execute all the tests in a Test Type without running Batch API

CALL CTRL_SP_RUN_TESTTYPE('U')

Test Results

The quickest way to check the results is to query the view

CTRL_V_PROCESS_TESTRESULT_LATEST.

It will show tests from your latest run and the status. The test query is also present in the view, it will help in investigating failed tests.

Test Results are stored for each execution along with Status and Run count in CTRL_T_PROCESS_TESTRESULT. Tests executed through Batch API has a Run_ID and others wont.

RUN_ID	PROCESS_TYPE_NAME	PROCESS_ID	GROUP_ID	TEST_ID	TEST_NAME	TEST_TYPE	RUN_COUNT	START_TIME	STOP_TIME	STATUS	EXPECTED_VALUE	ACTUAL_VALUE	MESSAGE
NULL	PARTY	230	25	2	UT_ADM_T_DIM_...	U	3	2017-05-02 12:35:24	2017-05-02 12:35:...	FAILURE	UNKNOWN		Actual value is N...
NULL	PARTY	230	25	2	UT_ADM_T_DIM_...	U	8	2017-05-03 09:07:14	2017-05-03 09:07:...	SUCCESS	UNKNOWN	UNKNOWN	Actual value is E...
3394	PARTY	230	25	1	UT_ADM_T_DIM_...	U	1	2017-05-02 12:33:52	2017-05-02 12:33:...	FAILURE	Individual		Actual value is N...
3394	PARTY	230	25	2	UT_ADM_T_DIM_...	U	1	2017-05-02 12:33:55	2017-05-02 12:33:...	SUCCESS	UNKNOWN	UNKNOWN	Actual value is E...
3395	PARTY	230	25	1	UT_ADM_T_DIM_...	U	5	2017-05-02 12:48:31	2017-05-02 12:48:...	SUCCESS	Individual	Individual	Actual value is E...
3395	PARTY	230	25	2	UT_ADM_T_DIM_...	U	6	2017-05-02 12:48:34	2017-05-02 12:48:...	SUCCESS	UNKNOWN	UNKNOWN	Actual value is E...
3396	PARTY	230	25	1	UT_ADM_T_DIM_...	A	1	2017-05-02 13:02:05	2017-05-02 13:03:...	SUCCESS	0	0	Actual value is E...
3396	PARTY	230	25	2	UT_ADM_T_DIM_...	A	1	2017-05-02 13:03:30	2017-05-02 13:04:...	SUCCESS	0	0	Actual value is E...

Grouping Test

Batch API can only identify two Test Types, "U" for Unit Test and "A" for Acceptance Test. But if the user want to create test which need to run outside Batch API like Release tests its possible by specifying TEST_TYPE to any single character like "R" or "S" in

CTRL_T_PROCESS_TESTS.

You can execute these special Test types by executing the stored procedure,CTRL_SP_RUN_TESTTYPE('R').

Generating Insert Statement

When creating Unit test we might need to create INSERT statements against specified tables. The following stored procedure will help to generate Insert statements for any WHERE condition.

CALL CTRL_SP_GENERATE_INSERT('**ADM_LND_D**','**LND_T_ADB_GEOGRAPHY_BRANCH**', '**SOURCE_CD = 01 AND BRCH_CD = 07**');

The above stored procedure call will create INSERT statement for the output of SELECT * FROM **ADM_LND_D..LND_T_ADB_GEOGRAPHY_BRANCH** WHERE **SOURCE_CD = 01 AND BRCH_CD = 07**