CA-1742798 WDS to Openlink Interface

.
Control Description
Daily, a Treasury Analyst runs the OpenLink (OL) Signature Search Results report to identify the transferred wire signers and confirms the status of the signer (i.e., terminated, transferred or retired) with the respective business unit management. The Analyst submits a request to Wilton Security Admin to remove the wire signer in OL. The final package with supporting documents (emails, screen prints, etc.) is reviewed and physically signed off by the Manager to ensure all changes have been actioned.
.
Further Description:
Wire approval authority for terminated employees is automatically deactivated by OpenLink based on the HR daily feed. Authorized wire signers that have transferred job functions per the HR feed are flagged in OpenLink. This test focuses on the uploading of WDS file to Openlink
Interface: WDS to Openlink
Usage: To automatically upload the daily WDS feed into OpenLink.
Applications: Openlink
.
Walk Through 1: 11/02/2017
Attendees: (IT) Cheryl Eco, Krystle Bautista, Aram Young, Ray Licata (PWC) Priyanka Tope (AARM) John Vinas (FCU) Jasmin Mathew, Allen Lum, Hoang Vinh Nguyen
.
Walk Through 2: 12/01/2017
Attendees: (IT) Cheryl Eco, Aram Young (PWC) Priyanka Tope (AARM) John Vinas (FCU) Allen Lum, Hoang Vinh Nguyen

---

TS-A : Determine the type of the report: a) standard b) customized c) query (with ITGC) d) ad-hoc query (without ITGC) e) other

Pg. 2

TS-B : Determine the completeness of the report

Pg. 28

TS-C : Determine the accuracy of the report

Pg. 28

TS-D : Determine the controls designed to assess the reliability of the key report (i.e., accuracy, completeness and reliability)

Pg. 2

No relevant exceptions noted.

TS-A : Determine the type of the report/interface: a) standard b) customized c) query (with ITGC) d) ad-hoc query (without ITGC) e) other ===> b) This is a customized/ automated interface
.

TS-A.01: FCU obtained from IT the workflow that uploads the WDS file to OL (pg. 3). FCU observed IT show the email with the daily status of the WDS upload (pg. 4)

TS-A.02: FCU obtained from IT the UNIX script "load_wds.ksh" that is used to upload the WDS file into the staging table (pp. 5-8)

TS-A.03: FCU obtained from IT the stored procedure "`OLF_EXT.PR_OL_AIG_WDS_HR`" that is used to upload the staging table into the HR Table (pp. 9-11)

TS-A.04: FCU obtained from IT the JAVA package "com.aig.ReferenceDataUpdate" that is used to update the OL Personnel data (pp. 12-26)

TS-A.05: FCU observed that the code is managed in TFS (pg. 27)

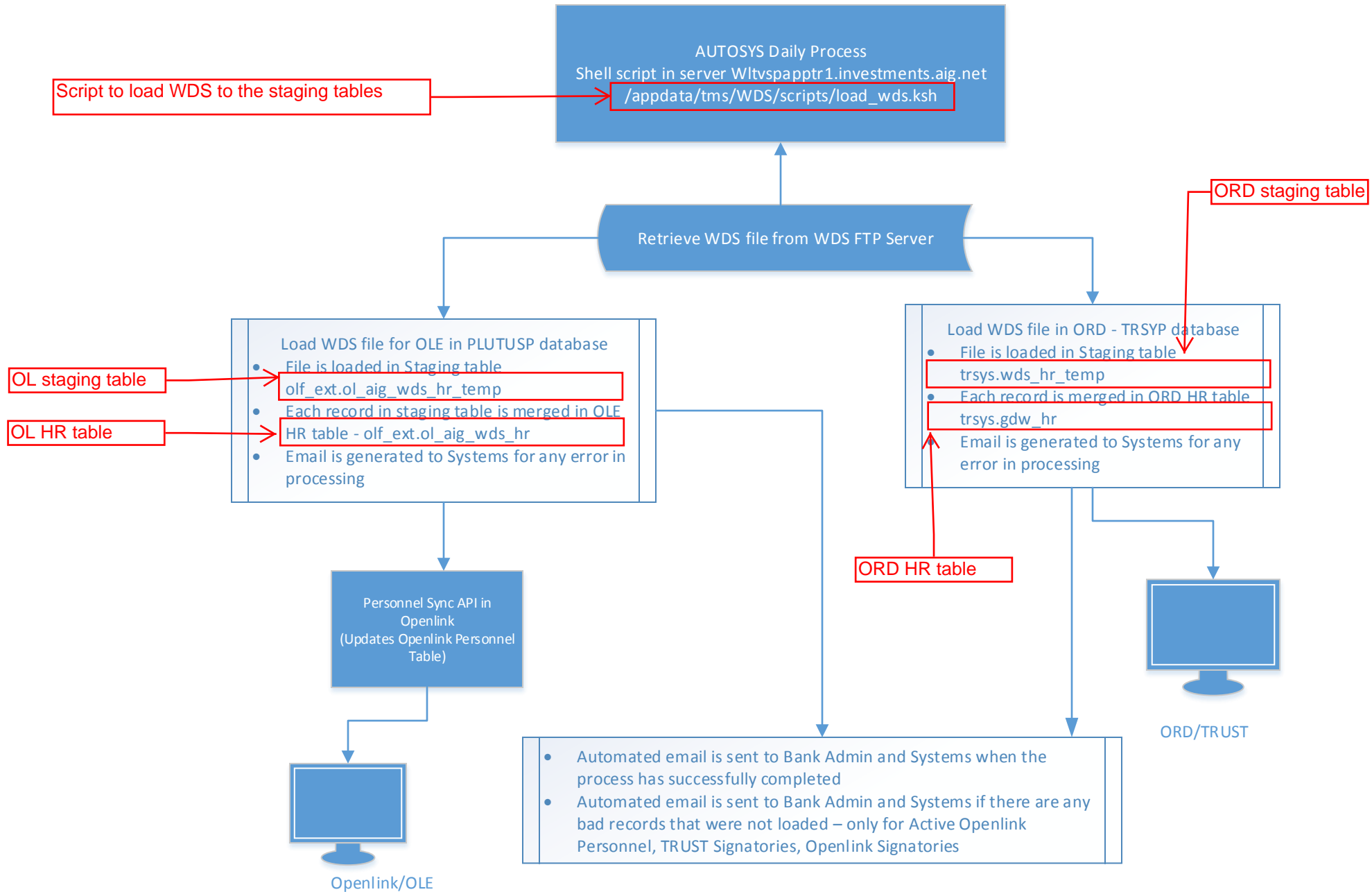The script, stored procedure and code are subject to IT Change Management

TS-D : Determine the controls designed to assess the reliability of the key report/interface (i.e., accuracy, completeness and reliability).
.

TS-D.01: Determine whether there is an initial user acceptance testing.

FCU was not able to obtain the initial user acceptance testing, since this report was built before 2017.

TS-D.02: Determine whether there is an ongoing user acceptance testing

Yes. Since the interface is subject to IT change management (TS-A.05), all changes to the report must go through the user acceptance testing phase.

"Type of Interface" TS-A.01

WDS Data Load Process – OLE and ORD

Script to load WDS to the staging tables

AUTOSYS Daily Process
Shell script in server Wltvspapptr1.investments.aig.net
/appdata/tms/WDS/scripts/load_wds.ksh

ORD staging table

Retrieve WDS file from WDS FTP Server

Load WDS file for OLE in PLUTUSP database
- File is loaded in Staging table
  olf_ext.ol_aig_wds_hr_temp
- Each record in staging table is merged in OLE
  HR table - olf_ext.ol_aig_wds_hr
- Email is generated to Systems for any error in
  processing

OL staging table

OL HR table

Load WDS file in ORD - TRSYP database
- File is loaded in Staging table
  trsys.wds_hr_temp
- Each record is merged in ORD HR table
  trsys.gdw_hr
- Email is generated to Systems for any
  error in processing

ORD HR table

Personnel Sync API in
Openlink
(Updates Openlink Personnel
Table)

ORD/TRUST

- Automated email is sent to Bank Admin and Systems when the
  process has successfully completed
- Automated email is sent to Bank Admin and Systems if there are any
  bad records that were not loaded – only for Active Openlink
  Personnel, TRUST Signatories, Openlink Signatories

Openlink/OLE

\\livpvnasamg02.investments.aig.net\honguyen$\My Documents\Visual Studio 2013\Projects\Script1.js   1

```ksh
#!/bin/ksh
# Script:        load_wds.ksh
#==============================================================================
#==============================================================================
# Standard Variables Section - start
#==============================================================================

export ROOT_DIR=/appdata/tms/WDS

export SCRIPT=`echo $0 | awk -F'/' '{ printf $NF }'`
export DATE=`date +%Y%m%d`
export DATE_LOG=`date +%Y%m%d%H%M`

export SCRIPTHOME=$ROOT_DIR/scripts
export INPUTDIR=$ROOT_DIR/inputs
export BADDIR=$ROOT_DIR/badfiles
export CTLDIR=$ROOT_DIR/ctl
export LOGDIR=$ROOT_DIR/logs
export ARCHIVED=$ROOT_DIR/archived
export CFG=/appdata/tms/cfg

export BADFILE=wds_hr_$DATE_LOG.bad

export DBNAME=plutusp
export ORACLE_USER=`grep $DBNAME $CFG/cfg.ini |awk '{print $2}'`
export ORACLE_PWD=`grep $DBNAME $CFG/cfg.ini | awk '{print $3}'`
export TNS_DB=`grep $DBNAME $CFG/cfg.ini |awk '{print $4}'`

export ORDDB=trsyp
export ORDDB_USER=`grep $ORDDB $CFG/cfg.ini |awk '{print $2}'`
export ORDDB_PWD=`grep $ORDDB $CFG/cfg.ini | awk '{print $3}'`
export ORDTNS_DB=`grep $ORDDB $CFG/cfg.ini |awk '{print $4}'`

export SFTP_SERVER=`grep -w wds_prod1 $CFG/cfg.ini |awk '{print $2}'`
export SFTP_USER=`grep -w wds_prod1 $CFG/cfg.ini |awk '{print $3}'`
export SFTP_PWD=`grep -w wds_prod1 $CFG/cfg.ini |awk '{print $4}'`
export SFTP_REMOTEDIR=`grep -w wds_prod1 $CFG/cfg.ini |awk '{print $5}'`

echo $SFTP_USER
echo $SFTP_USERPASS

export REC=`cat $CFG/gdw_email`
#export REC='cheryl.eco@aig.com'
#export REC='surgeon.devarapalli@aig.com'

#Oracle environment vars
#. $CFG/oracle_10.sh
#. $CFG/oracle_10.sh
. $CFG/oracle.sh

export SLF=$LOGDIR/load_wds.ksh.$DATE_LOG.log
umask 000
cd $SCRIPTHOME

rm $LOGDIR/load_wds.htm
touch $LOGDIR/load_wds.htm

exec > ${SLF} 2>&1
#==============================================================================
# Standard Variables Section - end
#==============================================================================

echo "Started processing for $0 at `date`."
echo "Started processing for $0 at `date`."  > $LOGDIR/load_wds.htm
```

```
echo "Started ftp for  $SFTP_SERVER at `date`."  >> $LOGDIR/load_wds.htm
echo "Inside FTP line $SFTP_SERVER"
#sftp $SFTP_USER@$SFTP_SERVER << EOF
#$SFTP_PWD
# lcd $INPUTDIR
# cd $SFTP_REMOTEDIR
#mget V_WDS*.dat
# quit
#EOF

#curl sftp://vmzldledmdstg12.r1-core.r1.aig.net/DOAWM/data/Outbound/DownStream/WJG/V_WDS_CURRENT_WJG_v1.0. ↙
    dat -k --user wdstrea:R56fGrti >> /appdata/tms/WDS/inputs/V_WDS_CURRENT_WJG_v1.0.dat

    curl sftp://$SFTP_SERVER/$SFTP_REMOTEDIR/V_WDS_CURRENT_WJG_v1.0.dat -k --user $SFTP_USER:$SFTP_PWD >> $ ↙
    INPUTDIR/V_WDS_CURRENT_WJG_v1.0.dat

    echo "Ended ftp at `date`."

if [ $? != 0 ]
    then
echo "Error:Loading WDS_HR at `date`."  >> $LOGDIR/load_wds.htm
mailx -s 'WDS HR LOAD ERROR' $REC < $LOGDIR/load_wds.htm
exit 1
fi
chmod 777 $INPUTDIR/V_WDS*.dat

rm  $INPUTDIR/wds_hr.dat
rm  $INPUTDIR/temp_wds_hr.dat

#Archive file before processing because it gets renamed
export STORAGE_FILE1=$INPUTDIR/emp_not_loaded.csv

echo "Archived the File which is FTP at `date`."

cp $INPUTDIR/V_WDS*.dat $ARCHIVED/V_WDS_CURRENT_WJG_v1.0.dat$DATE

mv $INPUTDIR/V_WDS*.dat $INPUTDIR/wds_hr.dat

chmod 777 $INPUTDIR/wds_hr.dat

#Check if there is a file to process before continuing

if [[ ! -f $INPUTDIR/wds_hr.dat ]] then
echo "Error:Loading WDS_HR at `date`. No valid file."  >> $LOGDIR/load_wds.htm
mailx -s 'WDS HR LOAD ERROR' $REC < $LOGDIR/load_wds.htm
exit 1
fi


# Below command removes First and last line in the file
#echo "Removes FIRST Line in the File at `date`."

#sed '1d' < $INPUTDIR/gds_hr.dat > $INPUTDIR/temp_gds_hr.dat
#chmod 777 $INPUTDIR/temp_gds_hr.dat

#if [ $? != 0 ]
#    then
#       echo "Error:Loading GDS_HR at `date`."  >> $LOGDIR/load_gds.htm
#       mailx -s 'GDS HR LOAD ERROR' $REC < $LOGDIR/load_gds.htm
#       exit 1
#fi

#mv $INPUTDIR/temp_gdw_hr.dat $INPUTDIR/gdw_hr.dat

#LOAD GDS_HR FILE
```

```
\\livpvnasamg02.investments.aig.net\honguyen$\My Documents\Visual Studio 2013\Projects\Script1.js          3

echo "Loading WDS_HR in OLE Database at `date`."  >> $LOGDIR/load_wds.htm

$ORACLE_BIN/sqlldr $ORACLE_USER/$ORACLE_PWD@$TNS_DB  control=$CTLDIR/wds_hr_temp.ctl data=$INPUTDIR/wds_hr. ↙
    dat log=$LOGDIR/wds_hr.log bad=$BADDIR/$BADFILE skip=1
#if [ $? != 0 ]
#   then
#       echo "Error:Loading GDS_HR in OLE DB at `date`."  >> $LOGDIR/load_gds.htm
#       mailx -s 'GDS HR LOAD ERROR in OLE DB' $REC < $LOGDIR/load_gds.htm
#       exit 1
#fi


echo "Running procedure for WDS_HR in OLE DB at `date`."  >> $LOGDIR/load_wds.htm          OLE
#Stored procedure for data load from Temp tables to original tables
sqlplus $ORACLE_USER/$ORACLE_PWD@$TNS_DB <<EOF
@exec_wds_hr_load.sql
EOF

if [ $? != 0 ]
   then
       echo "Error:Loading WDS_HR in OLE DB at `date`."  >> $LOGDIR/load_wds.htm
       mailx -s 'WDS HR LOAD ERROR in OLE DB' $REC < $LOGDIR/load_wds.htm
       exit 1
fi

echo "Loading WDS_HR in ORD Database at `date`."  >> $LOGDIR/load_wds.htm

$ORACLE_BIN/sqlldr $ORDDB_USER/$ORDDB_PWD@$ORDTNS_DB  control=$CTLDIR/ord_wds_hr.ctl data=$INPUTDIR/wds_hr. ↙
    dat log=$LOGDIR/ord_wds_hr.log bad=$BADDIR/ord_wds_hr_$DATE_LOG.bad skip=1
#if [ $? != 0 ]
#   then
#       echo "Error:Loading GDS_HR in ORD DB at `date`."  >> $LOGDIR/load_gds.htm
#       mailx -s 'GDS HR LOAD ERROR in ORD DB' $REC < $LOGDIR/load_gds.htm
#       exit 1
#fi


echo "Running procedure for WDS_HR in ORD DB at `date`."  >> $LOGDIR/load_wds.htm          ORD
#echo Database $ORDDB_USER $ORDDB_PWD $ORDTNS_DB
#Stored procedure for data load from Temp tables to original tables
sqlplus $ORDDB_USER/$ORDDB_PWD@$ORDTNS_DB <<EOF
@exec_ord_wds_hr_load.sql
EOF

if [ $? != 0 ]
   then
       echo "Error:Loading WDS_HR in ORD DB at `date`."  >> $LOGDIR/load_wds.htm
       mailx -s 'WDS HR LOAD ERROR in ORD DB' $REC < $LOGDIR/load_wds.htm
       exit 1
fi

echo "End of processing for $0 at `date`."  >> $LOGDIR/load_wds.htm

#mailx -s 'GDS HR LOAD success' $REC < $LOGDIR/load_gds.htm

if [[ ! -f $BADDIR/$BADFILE ]] then
     mailx -s 'WDS HR LOAD SUCCESS' $REC < $LOGDIR/load_wds.htm
else
$ORACLE_BIN/sqlldr $ORACLE_USER/$ORACLE_PWD@$TNS_DB  control=$CTLDIR/wds_hr_bad_data.ctl data=$BADDIR/$      ↙
    BADFILE log=$LOGDIR/wds_hr_bad_file.log bad=$BADDIR/wds_bad_file.bad skip=0
sqlplus $ORACLE_USER/$ORACLE_PWD@$TNS_DB <<EOF
@exec_wds_bad_file.sql $STORAGE_FILE1
EOF
filesize=`ls -ltr $STORAGE_FILE1 |awk '{print $5}'`
if [ $filesize -gt 53 ]
   then
```

```
    mailx -s 'WDS HR LOAD SUCCESS WITH BAD RECORDS' -a $STORAGE_FILE1 $REC < $LOGDIR/load_wds.htm
#     mailx -s 'WDS HR LOAD SUCCESS WITH BAD RECORDS' -a $BADDIR/$BADFILE $REC < $LOGDIR/load_wds.htm
else
    mailx -s 'WDS HR LOAD SUCCESS' $REC < $LOGDIR/load_wds.htm
fi

fi

exit 0
```

```
CREATE OR REPLACE PROCEDURE OLF_EXT.PR_OL_AIG_WDS_HR
AS
BEGIN
    MERGE INTO OLF_EXT.ol_aig_gdw_hr a
        USING (SELECT d.WKR_ENT_ID,
            d.WKR_TYPE_CD,
            d.WKR_FIRST_NM,
            d.WKR_LAST_NM,
            d.WKR_MID_NM,
            d.HIRE_DT,
            d.JOB_CD,
            d.JOB_TITLE,
            d.JOB_GRADE_NO,
            d.WORLD_WIDE_COMPANY_CD,
            d.GEMS_ID,
            d.BUSINESS_UNIT,
            d.REGION_CD,
            d.REGION_NM,
            d.SHELL_CD,
            d.SHELL_TX,
            d.DIVISION_CD,
            d.DIVISION_TX,
            d.COMPANY_CD,
            d.COMPANY_TX,
            d.BRANCH_CD,
            d.BRANCH_TX,
            d.DEPARTMENT_CD,
            d.DEPARTMENT_TX,
            d.DEPARTMENT_SERVICED,
            d.WKR_OFFC_CITY_TX,
            d.WKR_OFFC_STATE_PROV_TX,
            d.WKR_OFFC_COUNTRY_CD,
            d.CURRENCY_CD,
            d.WKR_STATUS_CD,
            d.LAST_WORKED_DT,
--                      d.LAST_DAY_PAID,
            d.TERMINATION_DT,
            d.FULL_PART_TIME_IN ,
            d.MANAGER_ENT_ID,
            d.MANAGER_FIRST_NM||' '||d.MANAGER_LAST_NM||' '||d.MANAGER_MID_NM as MANAGER_NAME,
            d.WKR_EMAIL_TX,
            d.WKR_OFFC_ADDR1_TX,
            d.WKR_OFFC_ADDR2_TX
                FROM OLF_EXT.ol_aig_wds_hr_TEMP d) b
            ON (a.EMPLOYEE_ID = b.WKR_ENT_ID)
    WHEN MATCHED
    THEN
        UPDATE SET EMPLOYEE_TYPE = b.WKR_TYPE_CD,
                EMP_FIRST_NAME = b.WKR_FIRST_NM ,
                EMP_LAST_NAME = b.WKR_LAST_NM,
                EMP_MID_NAME = b.WKR_MID_NM,
                HIRE_DATE = b.HIRE_DT,
                JOB_CODE = b.JOB_CD,
                JOB_TITLE = b.JOB_TITLE,
                GLOBAL_JOB_GRADE = b.JOB_GRADE_NO,
                WWCC = b.WORLD_WIDE_COMPANY_CD,
                GEMS_ID = b.GEMS_ID,
                BUSINESS_UNIT = b.BUSINESS_UNIT,
                REGION_CODE = b.REGION_CD,
                REGION_NAME = b.REGION_NM,
                SHELL_CODE = b.SHELL_CD,
                SHELL_DESC = b.SHELL_TX,
                DIVISION = b.DIVISION_CD,
                DIVISION_DESC = b.DIVISION_TX,
                COMPANY_NUM = b.COMPANY_CD,
                COMPANY_DESC = b.COMPANY_TX,
```

```
                        BRANCH_NUM = b.BRANCH_CD,
                        BRANCH_DESC = b.BRANCH_TX,
                        DEPT_NUM = b.DEPARTMENT_CD,
                        DEPT_DESC = b.DEPARTMENT_TX,
                        DEPT_SERVICED = b.DEPARTMENT_SERVICED,
                        WORK_CITY = b.WKR_OFFC_CITY_TX ,
                        WORK_STATE = b.WKR_OFFC_STATE_PROV_TX,
                        WORK_COUNTRY = b.WKR_OFFC_COUNTRY_CD,
                        CURRENCY_CODE = b.CURRENCY_CD,
                        EMP_STATUS = b.WKR_STATUS_CD,
                        LAST_DAY_WORK = b.LAST_WORKED_DT,
--                        LAST_DAY_PAID = b.LAST_DAY_PAID,
                        TERMINATION_DATE = b.TERMINATION_DT,
                        PART_FULL_INDICATOR = b.FULL_PART_TIME_IN,
                        MANAGER_ID = b.MANAGER_ENT_ID,
                        MANAGER_NAME = b.MANAGER_NAME,
                        EMAIL_ADDRESS = b.WKR_EMAIL_TX,
                        SITE = b.WKR_OFFC_ADDR1_TX,
                        FLOOR = b. WKR_OFFC_ADDR2_TX
    WHEN NOT MATCHED
    THEN
        INSERT      (EMPLOYEE_ID,
                     EMPLOYEE_TYPE,
                     EMP_FIRST_NAME,
                     EMP_LAST_NAME,
                     EMP_MID_NAME,
                     HIRE_DATE,
                     JOB_CODE,
                     JOB_TITLE,
                     GLOBAL_JOB_GRADE,
                     WWCC,
                     GEMS_ID,
                     BUSINESS_UNIT,
                     REGION_CODE,
                     REGION_NAME,
                     SHELL_CODE,
                     SHELL_DESC,
                     DIVISION,
                     DIVISION_DESC,
                     COMPANY_NUM,
                     COMPANY_DESC,
                     BRANCH_NUM,
                     BRANCH_DESC,
                     DEPT_NUM,
                     DEPT_DESC,
                     DEPT_SERVICED,
                     WORK_CITY,
                     WORK_STATE,
                     WORK_COUNTRY,
                     CURRENCY_CODE,
                     EMP_STATUS,
                     LAST_DAY_WORK,
--                      LAST_DAY_PAID,
                     TERMINATION_DATE,
                     PART_FULL_INDICATOR,
                     MANAGER_ID,
                     MANAGER_NAME,
                     EMAIL_ADDRESS,
                     SITE,
                     FLOOR)
        VALUES (b.WKR_ENT_ID,
            b.WKR_TYPE_CD,
            b.WKR_FIRST_NM,
            b.WKR_LAST_NM,
            b.WKR_MID_NM,
            b.HIRE_DT,
```

```
            b.JOB_CD,
            b.JOB_TITLE,
            b.JOB_GRADE_NO,
            b.WORLD_WIDE_COMPANY_CD,
            b.GEMS_ID,
            b.BUSINESS_UNIT,
            b.REGION_CD,
            b.REGION_NM,
            b.SHELL_CD,
            b.SHELL_TX,
            b.DIVISION_CD,
            b.DIVISION_TX,
            b.COMPANY_CD,
            b.COMPANY_TX,
            b.BRANCH_CD,
            b.BRANCH_TX,
            b.DEPARTMENT_CD,
            b.DEPARTMENT_TX,
            b.DEPARTMENT_SERVICED,
            b.WKR_OFFC_CITY_TX,
            b.WKR_OFFC_STATE_PROV_TX,
            b.WKR_OFFC_COUNTRY_CD,
            b.CURRENCY_CD,
            b.WKR_STATUS_CD,
            b.LAST_WORKED_DT,
--                      b.LAST_DAY_PAID,
            b.TERMINATION_DT,
            b.FULL_PART_TIME_IN ,
            b.MANAGER_ENT_ID,
            b.MANAGER_NAME,
            b.WKR_EMAIL_TX,
            b.WKR_OFFC_ADDR1_TX,
            b.WKR_OFFC_ADDR2_TX );
  COMMIT;
/* Below Delete Statement is been commented for System Issue 182 on 12272012 */
/* Delete from OLF_EXT.ol_aig_gdw_hr g
  where not exists (select 1 from OLF_EXT.ol_aig_gdw_hr_temp t
            where t.EMPLOYEE_ID = g.employee_id);
commit; */
EXCEPTION
   WHEN OTHERS
   THEN
      RAISE_APPLICATION_ERROR (
         -20001,
         'An error was encountered - ' || SQLCODE || ' -ERROR- ' || SQLERRM);
END PR_ol_aig_wds_hr;
/
```

H:\02 WebEx AIG Apps\2017-1012\2017-11-02 Treasury ...Invitation 2017-10-26\WDS to OL\PersonnelUpdate.java1

```java
/*
 *
 *  Jacob Florentin   06/20/2016    Parameterized LOG DIR PATH from USER_PLUGIN_ARGS
 *
 */

package com.aig.ReferenceDataUpdate;

import com.olf.cs.utility.plugin.PluginArgs;
import com.olf.openjvs.*;
import com.olf.openjvs.enums.*;
import com.olf.openjvs.fnd.RefBase;
import com.olf.openjvs.ODateTime;
import java.io.File;

public class PersonnelUpdate implements IScript {

    private final String sLog_Success = "SUCCESS";
    private final String sLog_Data = "DATA";
    private final String sLog_Error = "ERROR";
    private final String sLog_Ex = "EXCEPTION";
    private final String sLog_Warn = "WARNING";

    private Table tblPersonnel, tblAccountSig;
    private Table tblListPersonnelInfoTypes;
    private Table tblTestId;
    private Table tblLogTemp;

    private String sScriptName = "";
    /*
        Log folder Path should be as below and has been setup in user_plugin_args table. Script postifx      ↙
    with classname(PERSONNELUPDATE)
     */
    private String sLogDirPath = "";
    private boolean bApplyUpdates = true;

    private String sErrorFileName = "" ;
    private String sExceptionFileName = "" ;
    private String sStartMsg = "";

    // Flag to update OL, some times needed for log purpose only
    private int iRunId = 0, iScriptId = 0;

    public PersonnelUpdate() throws OException {

        sScriptName = this.getClass().getSimpleName();

        tblPersonnel = new Table("PersonneList");
        tblAccountSig = new Table("AccountSignatoryList");
        tblListPersonnelInfoTypes = new Table("PersonnelInfoTypesList");
        tblTestId = new Table("TestDataIdList");
        tblLogTemp = new Table("Log");
        tblLogTemp.addCols("I(RunId), S(LogType),S(LogMsg), S(LogMsg_1),S(LogMsg_2),S(LogTable),S(LogCol),  ↙
    I(DataId), S(OldVal),S(NewBal)");

        //Get the current server date & time and save them
        int iServerTime = 0, iServerDate;
        ODateTime OSaveSrvrDateTime = ODateTime.dtNew();
        iServerDate = OCalendar.getServerDate();
        iServerTime = Util.timeGetServerTime();
        OSaveSrvrDateTime.setDateTime(iServerDate, iServerTime);
        String sFileDateTime = Str.dtToString(OSaveSrvrDateTime, DATE_FORMAT.DATE_FORMAT_ISO8601.toInt(),  ↙
    TIME_FORMAT.TIME_FORMAT_ISO8601.toInt());

        Table tblInfo = Util.NULL_TABLE;
```

H:\02 WebEx AIG Apps\2017-1012\2017-11-02 Treasury ...Invitation 2017-10-26\WDS to OL\PersonnelUpdate.java2

```
        try {
            tblInfo = RefBase.getInfo();
            sStartMsg = String.format("Host Name = %s, Script Name = %s, Server = %s, DB = %s,        ↙
        bApplyUpdates = %s, Run ID = %s."
                    , tblInfo.getString("hostname", 1), tblInfo.getString("script_name", 1)
                    , tblInfo.getString("server", 1), tblInfo.getString("database", 1), bApplyUpdates,        ↙
        iRunId);

            PluginArgs myPluginArgs = new PluginArgs();
            myPluginArgs.load("PersonnelUpdate", "PERSONNEL_UPDATE");
            String sLogDirPathPROD=myPluginArgs.getValue("LOG_DIR_PATH_PROD");
            String sLogDirPathUAT=myPluginArgs.getValue("LOG_DIR_PATH_UAT");
            String sServer = myPluginArgs.getValue("PROD_DB_SERVER");

            if (tblInfo.getString("server", 1).contains(sServer) == true)
                sLogDirPath = sLogDirPathPROD;
            else
                sLogDirPath = sLogDirPathUAT;

        }
        catch (OException oe) {
            OLog.logInfo(2, "\n\nError = " + this.getClass().getSimpleName() + ": " + oe.getMessage());
        }
        finally {
            tblInfo.destroy();
        }
        if (Str.isEmpty(sLogDirPath) != 0)
        {
            Util.exitFail("An Error Occurred in " + sScriptName + ". No Output Path Properties Parameter in ↙
        USER_plugin_args.");
        }
        sLogDirPath = sLogDirPath + "\\" + sScriptName + "\\";
        sErrorFileName = sLogDirPath + sScriptName + "_ERROR_" + sFileDateTime + ".log";
        sExceptionFileName = sLogDirPath + sScriptName + "_EXCEPTION_" + sFileDateTime +".log";
    }

    @Override
    public void execute(IContainerContext context) throws OException {
        try {

            beginProcess();

            // populate table with list that needs to be updated in personnel and account_signatory
            populateTables();

            // update personnel
            updatePersonnel();

            // update account signatory
            updateAccountSignatory();

            // update FX Sig Mapppings
            updateFXSigMappings();

            Table tblException = new Table();
            tblException.select(tblLogTemp, "LogType"," LogType EQ " + sLog_Ex);
            // create exception log file only if exceptions exists
            if (tblException.getNumRows() > 0 )
            {
                File f = new File(sExceptionFileName);
                if(!f.exists())
                {
                    //Util.errorInitScriptErrorLog(sExceptionFileName);
                    Util.errorLogMessage(sExceptionFileName,"START",sStartMsg);
                    Util.errorLogMessage(sExceptionFileName,"WARNING", "There are Exception for Personnel/ ↙
        Account Signatory.");
```

H:\02 WebEx AIG Apps\2017-1012\2017-11-02 Treasury ...Invitation 2017-10-26\WDS to OL\PersonnelUpdate.java3

```java
                    Util.errorLogMessage(sExceptionFileName,"END","");
                }
            }
            Util.exitSucceed();
        } catch (OException e) {
            handleError(e,"");
            logErrorMessage("END");
            Util.exitFail();
        }
        finally{
            logToDB("END", "", "","","","", 0, "", "");
            // Clean up memory
            if (Table.isTableValid(tblPersonnel) == 1)
                tblPersonnel.destroy();
            if (Table.isTableValid(tblAccountSig) == 1)
                tblAccountSig.destroy();
            if (Table.isTableValid(tblListPersonnelInfoTypes) == 1)
                tblListPersonnelInfoTypes.destroy();
            if (Table.isTableValid(tblLogTemp) == 1)
                tblLogTemp.destroy();
        }
    }

    private void populateTables() throws OException
    {
        // get all the employees with authorized status or exceptions
        String sSql_1 = ""
                    + "\n   WITH "
                    + "\n   personnel_info_view AS ( "
                    + "\n       SELECT  p.personnel_id, p.type_id, p.info_value, p1.type_name, p1.        ↵
personnel_type "
                    + "\n        FROM    personnel_info p, personnel_info_types p1 "
                    + "\n        WHERE   p.type_id = p1.type_id "
                    + "\n    ) "
                    + "\n   , personnel_view AS ( "
                    + "\n       SELECT   p.id_number as personnel_id, p.* "
                    + "\n               , p1.info_value AS termination_date "

                    + "\n               , (CASE WHEN trim(hr.employee_id) is not null then 'Y' else 'N'   ↵
end) as hr_rec_found "
                    + "\n               , initcap(hr.emp_first_name) AS hr_first_name, initcap(hr.        ↵
emp_last_name) AS hr_last_name "
                    + "\n               , substr(hr.job_title,1,32) AS hr_title, hr.emp_status as hr_status ↵
, hr.email_address AS hr_email "
                    + "\n               , hr.termination_date AS hr_termination_date, hr.last_day_work AS  ↵
hr_last_day_work "
                    + "\n               , trunc(nvl(hr.last_day_work,hr.termination_date)) as              ↵
hr_end_date_orig "
                    + "\n               , to_char(nvl(hr.last_day_work,hr.termination_date),'mm/dd/yyyy')  ↵
AS hr_end_date "
                    + "\n        FROM    personnel p "
                    + "\n        left JOIN olf_ext.ol_aig_gdw_hr hr ON p.employee_id = hr.employee_id "
                    + "\n        LEFT JOIN personnel_info_view p1 ON p.id_number = p1.personnel_id "
                    + "\n             AND p.personnel_type = p1.personnel_type AND p1.type_name =          ↵
'Termination Date' "
                    + "\n    ) "
                    + "\n   , acc_sig AS ( "
                    + "\n       SELECT  p.id_number AS personnel_id, "
                    + "\n               nvl(count(*),0) as active_sig_count, '' as acc_sig_list"
                    + "\n               --listagg(a.account_name || '(' || a.account_id || ')',', ') within ↵
  group(order by 1) as acc_sig_list "
                    + "\n        from    personnel_view p, account_signatory s, account a "
                    + "\n        where   p.id_number = s.account_signatory "
                    + "\n                and s.account_id = a.account_id "
                    + "\n                and ((status != 2 AND hr_status not in ('A','T','U','D')) "
                    + "\n                    or (hr_status not IN ('T','U','D') AND hr_end_date IS NOT NULL ↵
```

H:\02 WebEx AIG Apps\2017-1012\2017-11-02 Treasury ...Invitation 2017-10-26\WDS to OL\PersonnelUpdate.java4

```
          ) "
                              + "\n                              or  (status = 2 )) "
                              + "\n                         and trunc(sysdate) between effective_start_date and     ↙
          effective_end_date "
                              + "\n        group by id_number "
                              + "\n      ) "
                              + "\n   , data_1 AS ( "
                              + "\n        SELECT  p.*, s.acc_sig_list, s.active_sig_count "
                              + "\n             , CASE WHEN "
                              + "\n                         hr_status IN ('T','U','D') "
                              + "\n                         AND hr_end_date IS NOT NULL "
                              + "\n                         AND hr_end_date != NVL(termination_date,'~') "
                              + "\n                       THEN 'Y' "
                              + "\n                       ELSE 'N' "
                              + "\n                    END AS end_date_changed "
                              + "\n             , CASE WHEN "
                              + "\n                         status != 2 "
                              + "\n                         AND hr_status IN ('T','U','D') "
                              + "\n                         AND hr_end_date IS NOT NULL "
                              + "\n                         AND trunc(sysdate) > hr_end_date_orig "
                              + "\n                       THEN 'Y' "
                              + "\n                       ELSE 'N' "
                              + "\n                    END AS status_changed "

                              + "\n                 --  exp_1 = HR recrd not found "
                              + "\n             , CASE WHEN hr_rec_found= 'N' then 'Employee ID Not found' ELSE ''  ↙
          END AS exp_1 "
                              + "\n                 --  exp_2 = HR status in beloe but end_date "
                              + "\n             , CASE WHEN hr_status not IN ('T','U','D') AND hr_end_date IS NOT    ↙
          NULL "
                              + "\n                       THEN 'Personnel: HR Status - Last day Discrepency' ELSE ''  ↙
          END AS exp_2 "
                              + "\n                 -- exp_3 = status = 0 is Auth Pending "
                              + "\n             , CASE WHEN status = 0 THEN 'Personnel: Auth Pending - No updates'  ↙
          ELSE '' END exp_3 "
                              + "\n                 -- exp_4 = Personnel other status = 2(Dot not use) has HR status    ↙
          other then below and has acctive signatory mappings "
                              + "\n             , CASE WHEN status != 2 AND hr_status not in ('A','T','U','D') AND  ↙
          active_sig_count > 0 "
                              + "\n                       THEN 'Account Signatory: Non-Active HR Employee Status'      ↙
          ELSE '' END AS exp_4 "
                              + "\n                 -- exp_5 = Personnel with status = 2 (Do not use) and has active   ↙
          signatoy mappings "
                              + "\n             , CASE WHEN status = 2  AND active_sig_count > 0 "
                              + "\n                       THEN 'Account Signatory: Non-Active Personnel Status' ELSE ↙
          '' END AS exp_5 "
                              + "\n                 -- exp_6 = Personnel with HR status in T,U,D but has no end_date "
                              + "\n             , CASE WHEN hr_status IN ('T','U','D') AND hr_end_date IS NULL "
                              + "\n                       THEN 'Personnel: No HR end date' ELSE '' END AS exp_6 "

                              + "\n        FROM    personnel_view p "
                              + "\n        LEFT JOIN acc_sig s ON p.personnel_id = s.personnel_id "
                              + "\n      ) "
                              + "\n    SELECT  p.* "
                              + "\n             , CASE WHEN exp_1 IS NULL THEN '' ELSE 'Employee ID not found in HR     ↙
          table' END AS exp_1_desc "
                              + "\n             , CASE WHEN exp_2 IS NULL THEN '' "
                              + "\n                       ELSE 'HR Status = '|| hr_status || ' , HR Last Day work date = ↙
          ' || hr_end_date "
                              + "\n                          || (CASE WHEN  active_sig_count = 0 THEN '' ELSE ' ,     ↙
          Account Signatory Mapping count = ' || active_sig_count END) "
                              + "\n                     END AS exp_2_desc "
                              + "\n             , CASE WHEN exp_3 IS NULL THEN '' ELSE 'Personnel Status = Auth Pending↙
           ' END AS exp_3_desc "
                              + "\n             , CASE WHEN exp_4 IS NULL THEN '' "
                              + "\n                       ELSE 'HR Status = '|| hr_status "
```

H:\02 WebEx AIG Apps\2017-1012\2017-11-02 Treasury ...Invitation 2017-10-26\WDS to OL\PersonnelUpdate.java5

```java
                    + "\n                              || (CASE WHEN  active_sig_count = 0 THEN '' ELSE ' ,      ↙
Account Signatory Mapping count = ' || active_sig_count END) "
                    + "\n              END AS exp_4_desc "
                    + "\n         , CASE WHEN exp_5 IS NULL THEN '' "
                    + "\n             ELSE 'OLF Personnel Status = ' || status "
                    + "\n                              || (CASE WHEN active_sig_count = 0 THEN '' ELSE ' ,      ↙
Account Signatory Mapping count = ' || active_sig_count END) "
                    + "\n              END AS exp_5_desc "
                    + "\n         , CASE WHEN exp_6 IS NULL THEN '' "
                    + "\n             ELSE 'OLF Personnel Status = ' || status || ', HR Status = ' ||↙
 hr_status "
                    + "\n              END AS exp_6_desc "

                    + "\n    FROM    data_1 p "
                    + "\n    where   0 = 0 "
                    + "\n            AND ((hr_rec_found = 'Y' and p.status = 1) "
                    + "\n                 OR (exp_1 IS NOT NULL OR exp_2 IS NOT NULL OR exp_3 IS NOT NULL↙
 "
                    + "\n                     OR exp_4 IS NOT NULL OR exp_5 IS NOT NULL OR exp_5 IS NOT  ↙
NULL)) "
                    + "\n            --and p.personnel_id in (20335)--311 ,20310,20058) 20011,20177,20003   ↙
and rownum < 130 "
                    + "\n    ";

    // get all the active signatories for authorized accounts and exceptions
    String sSql_2 = ""
                    + "\n   WITH "
                    + "\n   acc_list AS ( "
                    + "\n       select  a.*, case when b.linked_account_id is not null then 'N' else 'Y'   ↙
end as update_account "
                    + "\n              , b.account_status as auth_amend_status "
                    + "\n       from    account a "
                    + "\n       left join (select d.linked_account_id, d.account_status from account d    ↙
where account_status !=1) b "
                    + "\n              on a.account_id = b.linked_account_id "
                    + "\n       where a.account_status = 1 "
                    + "\n   ) "
                    + "\n   , acc_sig_view AS ( "
                    + "\n       SELECT  p.id_number as personnel_id, p.employee_id, p.status, a.         ↙
account_status, a.update_account, a.auth_amend_status "
                    + "\n              , s.account_id, s.account_signatory_id, s.effective_start_date, s. ↙
effective_end_date "
                    + "\n              , to_char(NVL(hr.last_day_work,hr.termination_date), 'mm/dd/yyyy') ↙
AS hr_end_date "
                    + "\n       FROM    personnel p "
                    + "\n       INNER JOIN olf_ext.ol_aig_gdw_hr hr "
                    + "\n           ON p.employee_id = hr.employee_id "
                    + "\n           AND p.status = 1 "
                    + "\n           AND emp_status IN ('T','U','D') "
                    + "\n           AND nvl(hr.last_day_work,hr.termination_date) IS NOT NULL "
                    + "\n           AND trunc(sysdate) > nvl(hr.last_day_work,hr.termination_date) "
                    + "\n       INNER JOIN account_signatory s "
                    + "\n           ON p.id_number = s.account_signatory "
                    + "\n           AND nvl(hr.last_day_work,hr.termination_date) > trunc(s.            ↙
effective_start_date) "
                    + "\n           AND  nvl(hr.last_day_work,hr.termination_date) < trunc(s.           ↙
effective_end_date) "
                    + "\n       INNER JOIN acc_list a "
                    + "\n           ON a.account_id = s.account_id "
                    + "\n   ) "
                    + "\n   SELECT  s.* "
                    + "\n   FROM    acc_sig_view s "
                    + "\n   WHERE   0 = 0 ";

    int iRetSQLVal = 0;
    try{
```

```java
            if (tblTestId.getNumRows() > 0 )
            {
                iRetSQLVal = DBase.runQueryWithIdList(sSql_1,tblTestId,"personnel_id",tblPersonnel);
                if (iRetSQLVal != OLF_RETURN_CODE.OLF_RETURN_SUCCEED.toInt())
                    throw new OException(sSql_1);

                iRetSQLVal = DBase.runQueryWithIdList(sSql_2,tblTestId,"personnel_id",tblAccountSig);
                if (iRetSQLVal != OLF_RETURN_CODE.OLF_RETURN_SUCCEED.toInt())
                    throw new OException(sSql_2);
            }
            else
            {
                iRetSQLVal = DBaseTable.execISql(tblPersonnel, sSql_1);
                if (iRetSQLVal != OLF_RETURN_CODE.OLF_RETURN_SUCCEED.toInt())
                    throw new OException(sSql_1);

                iRetSQLVal = DBaseTable.execISql(tblAccountSig, sSql_2);
                if (iRetSQLVal != OLF_RETURN_CODE.OLF_RETURN_SUCCEED.toInt())
                    throw new OException(sSql_2);
            }

            // get all the personnel_info_types as lookup, used while updating personnel_info
            sSql_1 =    "       SELECT  type_id, type_name, personnel_type "
                    + "\n        FROM    personnel_info_types "
                    + "\n        WHERE   type_name = 'Termination Date'";

            iRetSQLVal = DBaseTable.execISql(tblListPersonnelInfoTypes, sSql_1);
            if (iRetSQLVal != OLF_RETURN_CODE.OLF_RETURN_SUCCEED.toInt())
                throw new OException(sSql_1);

        } catch ( OException e)
        {
            throw new OException("\nError in populateTables module.\n" + e.getMessage());
        }
    }

    private void beginProcess() throws OException
    {
        Table tblScript = new Table();
        String sTestDataId = "", sSQL;
        int iRetSQLVal;

        try{
            sSQL = "SELECT script_id, update_flag, test_data_id from olf_ext.referencedata_update_list
where script_name = '" + sScriptName + "'";
            iRetSQLVal = DBase.runSqlFillTable(sSQL, tblScript);
            if (iRetSQLVal != OLF_RETURN_CODE.OLF_RETURN_SUCCEED.toInt())
                throw new OException(sSQL);

            if (tblScript.getNumRows() != 1)
                throw new OException(String.format(" No/Invalid data in referencedata_update_list for
script = %s, count = %s . \n", sScriptName, tblScript.getNumRows()) + sSQL);

            iScriptId = tblScript.getInt("script_id", 1);
            if (tblScript.getString("update_flag", 1).equals("N"))
                bApplyUpdates = false;

            sTestDataId = tblScript.getString("test_data_id", 1);
            if (Str.isEmpty(sTestDataId) == 0)
            {
                sSQL = String.format("select to_number(column_value) as personnel_id from table(CAST
(olf_ext.fn_split('%s') AS  olf_ext.SPLIT_TBL))",sTestDataId);
                iRetSQLVal = DBaseTable.execISql(tblTestId, sSQL);
                if (iRetSQLVal != OLF_RETURN_CODE.OLF_RETURN_SUCCEED.toInt())
                    throw new OException(sSQL);
            }
```

```java
            //debug
            // tblScript.viewTableForDebugging();
            tblScript.destroy();

            Table tblRunId = new Table();
            sSQL = "SELECT olf_ext.referencedata_update_run_seq.NEXTVAL AS nextid FROM DUAL";
            iRetSQLVal = DBase.runSqlFillTable(sSQL,tblRunId);
            if (iRetSQLVal != OLF_RETURN_CODE.OLF_RETURN_SUCCEED.toInt())
                    throw new OException(sSQL);

            iRunId = tblRunId.getInt("nextid", 1);
            // tblRunId.viewTableForDebugging();
            tblRunId.destroy();

            Table tblRunArgs = Table.tableNew();
            tblRunArgs.addCols("I(prunid), I(pscriptid),S(puserid)");

            int iRow = tblRunArgs.addRow();
            tblRunArgs.setInt("prunid", iRow, iRunId);
            tblRunArgs.setInt("pscriptid", iRow, iScriptId);
            tblRunArgs.setString("puserid", iRow, Ref.getUserName());
            // tblRunArgs.viewTableForDebugging();

            iRetSQLVal = DBase.runProc("USER_PR_REFDATA_UPDATE_RUN", tblRunArgs);
            if (iRetSQLVal != OLF_RETURN_CODE.OLF_RETURN_SUCCEED.toInt())
                throw new OException("Procedure = USER_PR_REFDATA_UPDATE_RUN");

            logToDB("START", sStartMsg + String.format("ErrorFile = %s, ExceptionFile = %s",sErrorFileName, ↙
    sExceptionFileName), "","", "", "", 0, "", "");

        } catch ( OException e)
        {
            throw new OException("\nError in beginProcess module.\n" + e.getMessage());
        }
    }

    private void handleError(Exception e, String sMsg) {

        try{
            // check if error log file exist
            File f = new File(sErrorFileName);
            if(!f.exists())
            {
                logErrorMessage("START", sStartMsg);
            }
        } catch (OException oe) {
        }

        if (e instanceof OException) {
            String sError = e.getMessage();
            try {
                String sDetails= "";
                for (StackTraceElement ste : e.getStackTrace()) {
                    sDetails = sDetails + " \n " + ste.toString();
                }
                if (Str.isEmpty(sMsg) == 0)
                    logErrorMessage("sMsg = " + sMsg);

                logErrorMessage("\nsError 1 = " + sError);
                logErrorMessage("\nError Stack = " + sDetails);
                logToDB(sLog_Error, sError, "", "","","", 0, "", "");
            } catch (OException oe) {
                // Can't do much else !!!
            }
            return;
        }
```

```java
    }

    private void logToDB(String sLogType, String sLogMsg, String sLogMsg_1, String sLogMsg_2
            , String sLogTable,String sLogCol, int iDataID, String sOldVal, String sNewVal) throws
    OException
    {
        try
        {
            if (iRunId > 0)
            {
                Table tblLog= Table.tableNew();
                tblLog.addCols("I(prunid), S(plog_type),S(plog_msg), S(plog_msg_1),S(plog_msg_2)S
    (ptablename),S(pcolname), I(pdataid), S(poldval),S(pnewval)");

                int iRow = tblLog.addRow();
                tblLog.setInt("prunid", iRow, iRunId);
                tblLog.setString("plog_type", iRow,sLogType);
                tblLog.setString("plog_msg", iRow,sLogMsg);
                tblLog.setString("plog_msg_1", iRow,sLogMsg_1);
                tblLog.setString("plog_msg_2", iRow,sLogMsg_2);
                tblLog.setString("ptablename", iRow,sLogTable);
                tblLog.setString("pcolname", iRow,sLogCol);
                tblLog.setInt("pdataid", iRow, iDataID);
                tblLog.setString("poldval", iRow, sOldVal);
                tblLog.setString("pnewval", iRow, sNewVal);

                DBase.runProc("USER_PR_REFDATA_UPDATE_LOG",tblLog);
            }

            int rowIndex = tblLogTemp.addRow();
            tblLogTemp.setInt("RunId", rowIndex, iRunId);
            tblLogTemp.setString("LogType", rowIndex, sLogType);
            tblLogTemp.setString("LogMsg", rowIndex, sLogMsg);
            tblLogTemp.setString("LogMsg_1", rowIndex, sLogMsg_1);
            tblLogTemp.setString("LogMsg_2", rowIndex, sLogMsg_2);
            tblLogTemp.setString("LogTable", rowIndex, sLogTable);
            tblLogTemp.setString("LogCol", rowIndex, sLogCol);
            tblLogTemp.setInt("DataId", rowIndex, iDataID);
            tblLogTemp.setString("OldVal", rowIndex, sOldVal);
            tblLogTemp.setString("NewBal", rowIndex, sNewVal);
        } catch ( OException e)
        {
            handleError(e, "Error in logToDB.");
        }
    }

    private void updatePersonnel() throws OException
    {
        //OLog.logInfo(2,"\n Personnel Recordcount = " + tblPersonnel.getNumRows());

        for (int iRow = 1; iRow <= tblPersonnel.getNumRows(); iRow++)
        {
            int iPersonnelId = 0, iPeronnelInfoTypeId = 0, iPersonnelType = 0, iRetVal = 0;
            String sHRTerminationDate = "", sOLDBUpdate = "", sEmployeeId = "";
            boolean bPersonnelChanged = false;

            Table tblRetPersonnelOrig = Table.tableNew("ret_personnel_orig");
            Table tblRetPersonnel = Table.tableNew("ret_personnel");
            Table tblRetPersonnelInfo = Table.tableNew("ret_personnel_info");
            Table tblPersonnelInfoTypes = Table.tableNew("personnel_info_types");

            try {
                // personnel details
                iPersonnelId = tblPersonnel.getInt("personnel_id", iRow);
                sEmployeeId = tblPersonnel.getString("employee_id", iRow);
                iPersonnelType = tblPersonnel.getInt("personnel_type", iRow);
```

H:\02 WebEx AIG Apps\2017-1012\2017-11-02 Treasury ...Invitation 2017-10-26\WDS to OL\PersonnelUpdate.java9

```java
                sHRTerminationDate = tblPersonnel.getString("hr_end_date", iRow);

                //OLog.logInfo(2, String.format("\n Personnel details = %s , personnel id = %s, HR rec      ↙
        found = %s ",iRow, iPersonnelId,tblPersonnel.getString("hr_rec_found", iRow)));

                // update only for Authorized users
                if (Str.equal(tblPersonnel.getString("hr_rec_found", iRow), "Y") == 1 &&
                        tblPersonnel.getInt("status", iRow) == PERSONNEL_STATUS_TYPE.                          ↙
        PERSONNEL_AUTHORIZED_STATUS.toInt())
                {
                    // retrieve OLF personnel table
                    tblRetPersonnel = RefBase.retrievePersonnel(iPersonnelId);
                    tblRetPersonnelOrig = tblRetPersonnel.copyTable();

                    // set only if the value is changed
                    if (Str.iEqual(tblPersonnel.getString("first_name", iRow), tblPersonnel.getString(       ↙
        "hr_first_name", iRow)) != 1)
                    {
                        bPersonnelChanged = true;
                        tblRetPersonnel.setString("first_name", 1, tblPersonnel.getString("hr_first_name",    ↙
        iRow));
                    }
                    if (Str.iEqual(tblPersonnel.getString("last_name", iRow), tblPersonnel.getString(         ↙
        "hr_last_name", iRow)) != 1)
                    {
                        bPersonnelChanged = true;
                        tblRetPersonnel.setString("last_name", 1, tblPersonnel.getString("hr_last_name",      ↙
        iRow));
                    }
                    if (Str.iEqual(tblPersonnel.getString("email", iRow), tblPersonnel.getString("hr_email"   ↙
        , iRow)) != 1)
                    {
                        bPersonnelChanged = true;
                        tblRetPersonnel.setString("email", 1, tblPersonnel.getString("hr_email", iRow));
                    }
                    if (Str.iEqual(tblPersonnel.getString("title", iRow), tblPersonnel.getString("hr_title"   ↙
        , iRow)) != 1)
                    {
                        bPersonnelChanged = true;
                        tblRetPersonnel.setString("title", 1, tblPersonnel.getString("hr_title", iRow));
                    }
                    if (Str.iEqual(tblPersonnel.getString("status_changed", iRow), "Y") == 1)
                    {
                        bPersonnelChanged = true;
                        tblRetPersonnel.setInt("status", 1, PERSONNEL_STATUS_TYPE.                            ↙
        PERSONNEL_SUSPENDED_STATUS.toInt());
                    }
                    if (Str.iEqual(tblPersonnel.getString("end_date_changed", iRow),"Y") == 1)
                    {
                        bPersonnelChanged = true;

                        // retrieve personnel info
                        tblRetPersonnelInfo = tblRetPersonnel.getTable("personnel_info", 1);

                        // Retrieve personnel info types for this personnel
                        tblPersonnelInfoTypes.select(tblListPersonnelInfoTypes,"type_id,type_name,           ↙
        personnel_type","personnel_type EQ " + iPersonnelType);

                        if (tblPersonnelInfoTypes.getNumRows() == 1)
                        {
                            iPeronnelInfoTypeId = tblPersonnelInfoTypes.getInt("type_id", 1);
                            int iInfoRow = tblRetPersonnelInfo.unsortedFindInt("type_id",                    ↙
        iPeronnelInfoTypeId);
                            String sPersonnelInfoTerminationDate = "";

                            if (iInfoRow == -1) {;
```

```java
                    int newRow = tblRetPersonnelInfo.addRow();
                    tblRetPersonnelInfo.setInt(1, newRow,iPeronnelInfoTypeId);
                    tblRetPersonnelInfo.setString(2, newRow,sHRTerminationDate);
                } else {
                    sPersonnelInfoTerminationDate = tblRetPersonnelInfo.getString(2, iInfoRow);
                    tblRetPersonnelInfo.setString(2, iInfoRow,sHRTerminationDate);
                }

                tblRetPersonnel.setTable("personnel_info", 1,tblRetPersonnelInfo);
                logToDB(sLog_Data
                        , String.format("PersonnelId ID = %s , Employee ID =  %s", iPersonnelId ↵
    ,sEmployeeId)
                        , ""
                        , ""
                        , tblRetPersonnelInfo.getTableName()
                        , "Termination Date"
                        , iPeronnelInfoTypeId
                        , sPersonnelInfoTerminationDate
                        , sHRTerminationDate);
            } else {
                // log
                logToDB(sLog_Ex
                        , String.format("Personel Info type not found for personnel type = %s  ↵
    for personnel_id = %s ",iPersonnelType,iPersonnelId)
                        , ""
                        , ""
                        , tblPersonnelInfoTypes.getTableName()
                        , ""
                        , iPeronnelInfoTypeId
                        , ""
                        , "");
                continue;
            }
        }

        // debug
        // tTemp.viewTableForDebugging();
        // tblRetPersonnel.viewTableForDebugging();
        // tblRetPersonnelInfo.viewTableForDebugging();

        // need below flag to update OL personnels
        tblRetPersonnel.addCol("save_name_address_flag", COL_TYPE_ENUM.COL_INT);
        tblRetPersonnel.addCol("save_personnel_info_flag",COL_TYPE_ENUM.COL_INT);

        tblRetPersonnel.setColValInt("save_name_address_flag",1);
        tblRetPersonnel.setColValInt("save_personnel_info_flag", 1);

        if (bApplyUpdates && bPersonnelChanged)
        {
            try {
                iRetVal = Ref.savePersonnel(tblRetPersonnel);

                if (iRetVal == OLF_RETURN_CODE.OLF_RETURN_SUCCEED.toInt())
                {
                    sOLDBUpdate = sLog_Success;
                }
                else
                {
                    sOLDBUpdate = sLog_Error;
                }
                logToDB(sOLDBUpdate, "", "", "", tblRetPersonnel.getTableName(), "",          ↵
    iPersonnelId, "", "");
            } catch (OException e) {
                String sTemp = String.format("\n Error updating personnel, Personnel id = %s,  ↵
    Employee ID = %s ", iPersonnelId, sEmployeeId);
                handleError(e,sTemp);
```

H:\02 WebEx AIG Apps\2017-1012\2017-11-02 Treasury ... 2017-10-26\WDS to OL\PersonnelUpdate.java   11

```
                            // if the user has not privilege it will still update the other section where    ↙
    user has privileges
                            Table tblRetPersonnelTemp = Table.tableNew();
                            tblRetPersonnelTemp = RefBase.retrievePersonnel(iPersonnelId);
                            tblRetPersonnel.setString("first_name", 1, tblRetPersonnelTemp.getString(       ↙
    "first_name", 1)) ;
                            tblRetPersonnel.setString("last_name", 1, tblRetPersonnelTemp.getString(        ↙
    "last_name", 1));
                            tblRetPersonnel.setString("email", 1, tblRetPersonnelTemp.getString("email",    ↙
    1));
                            tblRetPersonnel.setString("title", 1, tblRetPersonnelTemp.getString("title",    ↙
    1));
                            tblRetPersonnelTemp.destroy();
                    }
                }
                // remove extra columns added for diffTablesDetail in logDifference to function            ↙
    correctly.
                tblRetPersonnel.delCol("save_name_address_flag");
                tblRetPersonnel.delCol("save_personnel_info_flag");
                logDifference(tblRetPersonnelOrig, tblRetPersonnel," Employee ID = " + sEmployeeId);

            } //if (Str.equal(tblPersonnel.getString("hr_rec_found", iRow), "Y") == 1)

            //log exception
            for(int iExp = 1; iExp <= 6; iExp++)
            {
                String sExp_name = "exp_" + iExp;
                String sExp_Type = tblPersonnel.getString(sExp_name, iRow);
                if ( Str.isEmpty(sExp_Type) == 0)
                {
                    logToDB(sLog_Ex, sExp_Type,sEmployeeId, tblPersonnel.getString(sExp_name + "_desc",↙
     iRow), "", "", iPersonnelId, "", "");
                }
            }
        } catch (OException e) {
            String sTemp = String.format("\n Error updating personnel, Personnel id = %s, Employee ID = ↙
     %s ", iPersonnelId, sEmployeeId);
            handleError(e,sTemp);
        }

        // Clean up memory
        if (Table.isTableValid(tblRetPersonnelOrig) == 1)
            tblRetPersonnelOrig.destroy();
        if (Table.isTableValid(tblRetPersonnel) == 1)
            tblRetPersonnel.destroy();
        // comment below because error was thrown in error log not sure why
        //if (Table.isTableValid(tblRetPersonnelInfo) == 1) tblRetPersonnelInfo.destroy();
        if (Table.isTableValid(tblPersonnelInfoTypes) == 1)
            tblPersonnelInfoTypes.destroy();
    } // end of for loop
}

// update account signatory for terminated employees
private void updateAccountSignatory() throws OException
{
    int iRetVal = 0;
    Table tblAccounts = Table.tableNew("AccountList");

    tblAccounts.select(tblAccountSig, "DISTINCT, account_id, update_account, auth_amend_status ","     ↙
    account_id GT 0");
    // debug
    //tblAccountSig.viewTableForDebugging();
    //tblAccounts.viewTableForDebugging();

    for(int iAccRow=1; iAccRow <= tblAccounts.getNumRows();iAccRow++ )
    {
```

```java
        int iAccountId = 0;
        try{
            iAccountId = tblAccounts.getInt("account_id",iAccRow);

            Table tblPersonnelTerminated = Table.tableNew("PersonnelTerminated");
            tblPersonnelTerminated.select(tblAccountSig, "personnel_id, employee_id,
    account_signatory_id, hr_end_date "," account_id EQ " + iAccountId);

            if (Str.equal(tblAccounts.getString("update_account",iAccRow), "Y") == 1)
            {
                Table tblRetAccount = Table.tableNew("RetAccount");
                Table tblRetAccountSignatoryOrig = Table.tableNew("RetAccountSignatoryOrig");
                Table tblRetAccountSignatory = Table.tableNew("RetAccountSignatory");
                // retrieve accounts table
                tblRetAccount = Ref.retrieveAccount(iAccountId);

                // retrieve account signatory table
                tblRetAccountSignatory = tblRetAccount.getTable("account_signatory",1);
                tblRetAccountSignatoryOrig = tblRetAccountSignatory.copyTable();

                // debug
                //tblPersonnelTerminated.viewTableForDebugging();

                // loop thru each sig mapping
                for(int iPersonnelTerRow=1; iPersonnelTerRow <= tblPersonnelTerminated.getNumRows();
        iPersonnelTerRow++ )
                {
                    int iAccountSigRowIndex = 0, iOldEffectiveEndDate, iNewEffectiveEndDate;
                    String sOldEffectiveEndDate = "", sNewEffectiveEndDate = "";

                    sNewEffectiveEndDate = tblPersonnelTerminated.getString("hr_end_date",
        iPersonnelTerRow);
                    iNewEffectiveEndDate = OCalendar.parseString(sNewEffectiveEndDate);

                    iAccountSigRowIndex = tblRetAccountSignatory.unsortedFindInt("account_signatory_id"
        ,tblPersonnelTerminated.getInt("account_signatory_id", iPersonnelTerRow));

                    //OLog.logInfo(3, String.format("\n Personnel Id = %s, HR End Date = %s (%s),
        Account Signatory Row Index = %s ", tblPersonnelTerminated.getInt("personnel_id", iPersonnelTerRow),
        iNewEffectiveEndDate,sNewEffectiveEndDate, iAccountSigRowIndex));

                    if (iAccountSigRowIndex > 0)
                    {
                        iOldEffectiveEndDate = tblRetAccountSignatoryOrig.getInt("effective_end_date",
        iAccountSigRowIndex);
                        ODateTime dOldEffectiveEndDate = ODateTime.dtNew();
                        dOldEffectiveEndDate.setDate(iOldEffectiveEndDate);
                        // 2958463 is 12/31/9999 which is blank in OL GUI
                        if (iOldEffectiveEndDate != 2958463)
                            sOldEffectiveEndDate = Str.dtToString(dOldEffectiveEndDate,DATE_FORMAT.
        DATE_FORMAT_MDY_SLASH.toInt(),0);

                        // set the new value
                        tblRetAccountSignatory.setDateTimeByParts("effective_end_date",
        iAccountSigRowIndex, iNewEffectiveEndDate,0);
                        //OLog.logInfo(3, String.format("\n old effective_end_date = %s (%s) ",
        iOldEffectiveEndDate, sOldEffectiveEndDate));

                        logToDB(sLog_Data
                            , " Account Id = " + iAccountId + ", Personnel ID = " +
        tblRetAccountSignatoryOrig.getInt("account_signatory",iAccountSigRowIndex)
                            , ""
                            , ""
                            ,tblRetAccountSignatory.getTableName(),"effective_end_date"
                            , tblRetAccountSignatoryOrig.getInt("account_signatory_id",
        iAccountSigRowIndex)
```

```java
                                        , sOldEffectiveEndDate
                                        , sNewEffectiveEndDate);
                            }
                            else
                            {
                                logToDB(sLog_Error
                                        , String.format("Account Signaory row Index NOT found, Account Id = %s, ⤶
   Personnel ID = %s ",iAccountId, tblRetAccountSignatoryOrig.getInt("account_signatory",    ⤶
iAccountSigRowIndex))
                                        , ""
                                        ,
                                        , tblRetAccountSignatory.getTableName()
                                        ,"effective_end_date"
                                        , 0
                                        , sOldEffectiveEndDate
                                        , sNewEffectiveEndDate);
                            }
                        }
                        tblRetAccount.setTable("account_signatory", 1, tblRetAccountSignatory);

                        String sOLDBUpdate = sLog_Success;
                        if ( bApplyUpdates)
                        {
                            iRetVal = Ref.updateAccount(tblRetAccount);
                            if(iRetVal != OLF_RETURN_CODE.OLF_RETURN_SUCCEED.toInt())
                            {
                                sOLDBUpdate = sLog_Error;
                            }
                            //log and go to next record
                            logToDB(sOLDBUpdate,"","", "",tblRetAccount.getTableName(),"",iAccountId,"","");
                        }
                        //debug
                        //tblRetAccount.viewTableForDebugging();
                        //tblRetAccountSignatory.viewTableForDebugging();
                        //tblRetAccountSignatoryOrig.viewTableForDebugging();

                        //Clean up memory
                        if (Table.isTableValid(tblRetAccount) == 1)
                            tblRetAccount.destroy();
                        // comment below because error was thrown in error log not sure why
                        //if (Table.isTableValid(tblRetAccountSignatory) == 1) tblRetAccountSignatory.destroy()⤶
;
                        if (Table.isTableValid(tblRetAccountSignatoryOrig) == 1)
                            tblRetAccountSignatoryOrig.destroy();
                    }
                    else  // no updates but exceptions
                    {
                        // log exception for sig mapping for accounts that are not authorized
                        for(int iPersonnelTerRow=1; iPersonnelTerRow <= tblPersonnelTerminated.getNumRows();   ⤶
iPersonnelTerRow++ )
                        {
                            logToDB(sLog_Ex, "Personnel: Account Not Authorized",tblPersonnelTerminated.      ⤶
getString("employee_id",iPersonnelTerRow)
                                    , " Account Id = " + iAccountId + ", Account Status = " + tblAccounts.    ⤶
getInt("auth_amend_status",iAccRow) + ". No Updates made to account signatory."
                                    , "",   "",tblPersonnelTerminated.getInt("personnel_id",iPersonnelTerRow), ⤶
"", "");
                        }
                    }
                    //Clean up memory
                    if (Table.isTableValid(tblPersonnelTerminated) == 1)
                        tblPersonnelTerminated.destroy();

            } catch(Exception ex) {
                handleError(ex,"Error updating account signatory.");
            }
```

```java
        }   // end of for loop
        //Clean up memory
        if (Table.isTableValid(tblAccounts) == 1)
            tblAccounts.destroy();
    }

    private void updateFXSigMappings() throws OException{
        try{
            String sSql_1 = ""
                        + "\n  MERGE INTO olf_ext.fx_account_signatory t"
                        + "\n  USING"
                        + "\n  ("
                        + "\n      WITH"
                        + "\n      acc_list AS ("
                        + "\n          select  a.*, case when b.linked_account_id is not null then 'N'
 else 'Y' end as update_account"
                        + "\n                  , b.account_status as auth_amend_status"
                        + "\n          from    olf.account a "
                        + "\n          left join (select d.linked_account_id, d.account_status from
olf.account d where account_status !=1) b"
                        + "\n                  on a.account_id = b.linked_account_id"
                        + "\n          where a.account_status = 1"
                        + "\n          )  "
                        + "\n      , acc_sig_view AS ( "
                        + "\n          SELECT  p.id_number as personnel_id, p.employee_id, p.status, a
.account_status, a.update_account, a.auth_amend_status"
                        + "\n                  , s.account_id,  s.effective_start_date, s.
effective_end_date "
                        + "\n                  , (NVL(hr.last_day_work,hr.termination_date)) AS
hr_end_date"
                        + "\n                  , s.ASSIGNMENT_ID, hr.termination_date,hr.last_day_work"
                        + "\n                  , case when  nvl(hr.last_day_work,hr.termination_date) >
trunc(s.effective_start_date) then 1 else 0 end as flag"
                        + "\n          FROM    olf.personnel p "
                        + "\n          INNER JOIN olf_ext.ol_aig_gdw_hr hr "
                        + "\n              ON p.employee_id = hr.employee_id"
                        + "\n              AND emp_status IN ('T','U','D') "
                        + "\n              AND nvl(hr.last_day_work,hr.termination_date) IS NOT NULL
 "
                        + "\n              AND trunc(sysdate) > nvl(hr.last_day_work,hr.
termination_date)"
                        + "\n          INNER JOIN  olf_ext.fx_account_signatory s "
                        + "\n                  ON p.id_number = s.PERSONNEL_ID "
                        + "\n              AND ("
                        //              when start date is prior to hr end date
                        + "\n                  (nvl(hr.last_day_work,hr.termination_date) < trunc
(s.effective_start_date)) "
                        //                  when hr end_date is between start and end date
                        + "\n                  or (nvl(hr.last_day_work,hr.termination_date) >=
trunc(s.effective_start_date) and nvl(hr.last_day_work,hr.termination_date) < trunc(s.
effective_end_date))"
                        + "\n                      )"
                        + "\n              INNER JOIN acc_list a ON a.account_id = s.account_id"
                        + "\n          )"
                        + "\n      SELECT  *  FROM acc_sig_view s1"
                        + "\n  ) s"
                        + "\n  on (t.PERSONNEL_ID =  s.personnel_id and t.ASSIGNMENT_ID = s.ASSIGNMENT_ID)"
                        + "\n  when matched then update set effective_start_date=(case when
effective_start_date > hr_end_date then hr_end_date else effective_start_date end),"
                        + "\n                  effective_end_date=hr_end_date,last_update_date
=sysdate,ACTION_CODE='U',UPD_PERSONNEL_ID=" + RefBase.getUserId();

            int iRetSQLVal = 0;
            iRetSQLVal = DBase.runSql(sSql_1);
            if (iRetSQLVal != OLF_RETURN_CODE.OLF_RETURN_SUCCEED.toInt())
                throw new OException(sSql_1);
```

```java
            logToDB(sLog_Success, "FX Signatory Mapping: Update Effective End date for terminated employees↙
", "", "", "fx_account_signatory","", 0, "", "");
        } catch (OException e)
        {
            throw new OException("\nError in beginProcess module.\n" + e.getMessage());
        }
    }

    private void logDifference(Table tblOrig, Table tblUpdated, String sLogMsg)throws OException
    {
        Table tblDiff = Table.tableNew();
        tblOrig.diffTablesDetail(tblUpdated, tblDiff);

        for (int iDiffRow = 1; iDiffRow <= tblDiff.getNumRows(); iDiffRow++) {
            logToDB(sLog_Data, sLogMsg, "", "", tblOrig.getTableName(),
                    tblDiff.getString("col_name", iDiffRow),
                    tblOrig.getInt(1, tblDiff.getInt("row", iDiffRow)),
                    tblDiff.getString("val1", iDiffRow),
                    tblDiff.getString("val2", iDiffRow));
        }
    }

    private void logErrorMessage(String sMessage) throws OException {
        //Util.errorWriteString(sLogDirPath + sErrorFileName, sMessage + "\n");
        Util.errorLogMessage(sErrorFileName, "ERROR", sMessage);
        return;
    }

    private void logErrorMessage(String sLogCode, String sMessage) throws OException {
        //Util.errorWriteString(sLogDirPath + sErrorFileName, sMessage + "\n");
        Util.errorLogMessage(sErrorFileName, sLogCode, sMessage);
        return;
    }
}
```

TS-B : Determine the completeness of the report/interface

.

TS-B.01: FCU watched IT verify the header of the WDS file for 2017-12-01 ("V_WDS_CURRENT_WJG_v1.0.dat20171201: There are 154,804 records in the file, including the header (Pg. 29).

TS-B.02: FCU watched IT display the log of the uploading operation as recorded in "load_wds.ksh.201712010759.log": there are 154,803 records + 1 header found in the file (Pg. 14)

TS-B.03: FCU watched IT display from the log the results of the uploading: Total records uploaded to the OL staging table = 154,803 + 1 header record (Pg. 32)

TS-B.04: FCU watched IT verify that the count of records in OL staging table = 154,803 (pg. 33)

TS-B.05: FCU watched IT verify that the count of records in OL HR table = 154,803 (pg. 34)

TS-B.06: FCU observed in Pg. 40: Of the 4 records with changes from 2017-12-01 WDS, only 1 record has status = "Terminated". FCU asked IT to verify that the "Terminated" staff from the 2017-12-01 WDS file is not a signatory in OL (Pg. 44)

.

The report/interface is <u>complete</u>.

TS-C : Determine the accuracy of the report/interface

.

Steps TS-B.01, TS-B.02

.

TS-C.01: FCU watched IT search record in WDS. FCU identified 10 fields (pg. 35): (1) WKR_ENT_ID = 1401539 (2) AD_Domain = R1-CORE (3) Work-office = 1000186 (4) Hire_Date = 2012-11-19 (5) Job_CP = FAGA0044 (6) Last_Update = 2017-12-01 07:15:41 (7) GEMS_ID = 23455 (8) Terminate_Date = 2017-11-29 (9) Last_Worked_Date = 2017-11-29 (10) Manager_ID = 5033824

TS-C.02: FCU watched IT query from the staging table "OLF_EXT.OL.AIG_WDS_HR_TEMP" (pg. 36). FCU obtained the results of the query (pg. 39). FCU verified that the 10 fields in the staging table match those in WDS. The uploading from WDS to the staging table is accurate.

TS-C.03: FCU watched IT query from the HR table "OLF_EXT.OL.AIG_GDW_HR" (pg. 37). FCU obtained the results of the query (pg. 40). FCU verified that 7 fields in the HR table match those in WDS: (1) WKR_ENT_ID = 1401539 (3) Work-office = 1000186 (4) Hire_Date = 2012-11-19 (7) GEMS_ID = 23455 (8) Terminate_Date = 2017-11-29 (9) Last_Worked_Date = 2017-11-29 (10) Manager_ID = 5033824. The updating of the HR table is accurate.

TS-C.04: FCU watched IT query the table "olf.Personnel_history", which contains the update history of the employee record (pg. 38, 41)

TS-C.05: FCU obtained the screenshot of WKR_ENT_ID = 1401539, PERSONNEL_D = 24155 (pg. 42, 43). FCU verified that the employee's status = TERMINATED

TS-C.06: FCU obtained the Openlink Signatory Search Result on 2017-12-01, which shows the signatories HR status. FCU verified that the above personnel is not in the search result.

.

The report/interface is <u>accurate</u>.

.

```
wltvspapptr1.investments.aig.net - PuTTY

-rwxrwxrwx. 1 tmsprd    tms    71310849 Nov 28 07:59 V_WDS_CURRENT_WJG_v1.0.dat201
71128
-rwxrwxrwx. 1 tmsprd    tms    71321893 Nov 29 07:59 V_WDS_CURRENT_WJG_v1.0.dat201
71129
-rwxrwxrwx. 1 tmsprd    tms    71355583 Nov 30 07:59 V_WDS_CURRENT_WJG_v1.0.dat201
71130
-rwxrwxrwx. 1 tmsprd    tms    71387115 Dec  1 07:59 V_WDS_CURRENT_WJG_v1.0.dat201
71201
-bash-4.1$ wc -l V_WDS_CURRENT_WJG_v1.0.dat20171201
154804 V_WDS_CURRENT_WJG_v1.0.dat20171201
-bash-4.1$ head -3 V_WDS_CURRENT_WJG_v1.0.dat20171201
WKR_ENT_ID|WKR_LAST_NM|WKR_FIRST_NM|WKR_MID_NM|AD_DOMAIN_TX|WKR_LAN_ID|WKR_EMAIL_TX|AIG_FAX_NO|AIG_MOBILE_NO|CBD|CO
ST_CENTER_CD|WKR_OFFC_CD|HIRE_DT|WKR_TYPE_CD|HOME_ADDR_1_TX|HOME_CITY_TX|HOME_OFFICE_LOC_TX|HOME_POSTAL_CD_TX|HR_CO
MPANY_CD|LOCALE_CD|WKR_OFFC_ADDR1_TX|WKR_OFFC_ADDR2_TX|WKR_OFFC_STATE_PROV_CD|WKR_OFFICE_POSTAL_CD|FULL_PART_TIME_I
N|PERS_EMAIL_ADDR_TX|PERS_FAX_NO|PERS_MOBILE_NO|PERS_PHONE_NO|REMOTE_WKR_IN|MANAGER_ENT_ID|TIMEZONE_CD|HOME_ADDR_2_
TX|TERMINATION_DT|WKR_STATUS_CD|CREATE_TS|AIG_PHONE_NO|WKR_OFFC_ADDR3_TX|HOME_COUNTY_TX|HOME_STATE_PROV_CD|HOME_CNT
RY_CD|WKR_OFFC_CITY_TX|HOME_ADDR_3_TX|HOME_COUNTRY_TX|LAST_WORKED_DT|LAST_UPDATE_TS|ACCTG_CD_TYPE_ID|WKR_OFFC_CNTY|
WKR_OFFC_COUNTRY_CD|WKR_OFFC_COUNTRY_TX|WKR_OFFC_BLDG_TX|WKR_OFFC_SPACE_TX|WKR_OFFC_STATE_PROV_TX|MAIL_STOP|JOB_CD|
JOB_GRADE_NO|JOB_TITLE|JOB_FAMILY|JOB_FUNCTION|WORLD_WIDE_COMPANY_CD|GEMS_ID|BUSINESS_UNIT|SHELL_CD|DIVISION_CD|COM
PANY_TX|BRANCH_TX|DEPARTMENT_TX|DEPARTMENT_SERVICED|CURRENCY_CD|MANAGER_FIRST_NM|MANAGER_LAST_NM|MANAGER_MID_NM|MAN
AGER_EMAIL_TX|REGION_CD|REGION_NM|SHELL_TX|DIVISION_TX|SAP_COST_CENTER|SAP_COMPANY_CODE|COMPANY_CD|BRANCH_CD|DEPART
MENT_CD|BUSINESS_TITLE|WKR_OFFC_COUNTRY_ISO3_CD|SERVICE_DT|WKR_ACCESS_STATUS_CD|LATTD|LNGTD|EMPL_STAT_RSN|SOURCE_NM
|EFORM_NUMBER|FIELDGLASS_NUMBER|VENDOR_NAME1|WORKER_CLASS
1478558|Nagatoishi|Shigeru|||||||2MDO||2017-07-14|S|||||||||||P|||||6510422|||2018-06-30|A|2017-07-14 03:30:07|||
|||||||2017-11-19 07:24:27|||JP|JAPAN|||||||||2029|48233||03520800|||||||Yuko|Itakura||itakuray@aig.co.jp|||AMERIC
AN HOME ASSURANCE - JAPAN||||||||JPN||A|||HIRED|NES|||DIVESTITURE: FUJI LIFE|DIVESTED WORKER
1478570|Tanoue|Kazuko|||R6-CORE|llv0961|||||2MDO||2017-07-14|S|||||||||||P|||||6510422|||2018-06-30|A|2017-07-21 03
:15:39|||||||||||2017-11-19 07:24:27|||JP|JAPAN|||||||||2029|48233||03520800|||||||Yuko|Itakura||itakuray@aig.co.jp
|||AMERICAN HOME ASSURANCE - JAPAN||||||||JPN||A|||HIRED|NES|||DIVESTITURE: FUJI LIFE|DIVESTED WORKER
-bash-4.1$
```

CA-1742798 WDS to OL-S01.29

"Completeness" TS-B.01 There are 154,804 records in the WDS file (including the header record)

```
wltvspapptr1.investments.aig.net - PuTTY

Started processing for /appdata/tms/WDS/scripts/load_wds.ksh at Fri Dec  1 07:59:02 EST 20.. CR-1742798 WDS to OL-S01.30
Inside FTP line livvspsftp02.investments.aig.net
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 68.0M  100 68.0M    0      0  49.2M      0  0:00:01  0:00:01 --:--:-- 49.2M
Ended ftp at Fri Dec  1 07:59:03 EST 2017.
rm: cannot remove `/appdata/tms/WDS/inputs/temp_wds_hr.dat': No such file or directory
Archived the File which is FTP at Fri Dec  1 07:59:03 EST 2017.

SQL*Loader: Release 11.2.0.3.0 - Production on Fri Dec 1 07:59:03 2017

Copyright (c) 1982, 2011, Oracle and/or its affiliates.  All rights reserved.

Commit point reached - logical record count 12
Commit point reached - logical record count 24
Commit point reached - logical record count 36
Commit point reached - logical record count 48
Commit point reached - logical record count 60
Commit point reached - logical record count 72
Commit point reached - logical record count 84
Commit point reached - logical record count 96
Commit point reached - logical record count 108
Commit point reached - logical record count 120
Commit point reached - logical record count 132
Commit point reached - logical record count 144
Commit point reached - logical record count 156
Commit point reached - logical record count 168
Commit point reached - logical record count 180
Commit point reached - logical record count 192
--More--(0%)
```

"Completeness" TS-B.02 - Log of the uploading as recorded in "load_wds.ksh.201712010759.log"

CA-1742798 WDS to OL-S01.31

```
Commit point reached - logical record count 154579
Commit point reached - logical record count 154591
Commit point reached - logical record count 154603
Commit point reached - logical record count 154615
Commit point reached - logical record count 154627
Commit point reached - logical record count 154639
Commit point reached - logical record count 154651
Commit point reached - logical record count 154663
Commit point reached - logical record count 154675
Commit point reached - logical record count 154687
Commit point reached - logical record count 154699
Commit point reached - logical record count 154711
Commit point reached - logical record count 154723
Commit point reached - logical record count 154735
Commit point reached - logical record count 154747
Commit point reached - logical record count 154759
Commit point reached - logical record count 154771
Commit point reached - logical record count 154783
Commit point reached - logical record count 154795
Commit point reached - logical record count 154803
```

"Completeness" TS-B.02 - Log of the uploading = Total count = 154,803 + 1 header record

```
SQL*Plus: Release 11.2.0.3.0 Production on Fri Dec 1 07:59:31 2017

Copyright (c) 1982, 2011, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.4.0 - 64bit Production
With the Partitioning, Real Application Clusters, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
```

```
DIVISION_TX                             NEXT      *  |      CHARACTER
SAP_COST_CENTER                         NEXT      *  |      CHARACTER
SAP_COMPANY_CODE                        NEXT      *  |      CHARACTER
COMPANY_CD                              NEXT      *  |      CHARACTER
BRANCH_CD                               NEXT      *  |      CHARACTER
DEPARTMENT_CD                           NEXT      *  |      CHARACTER

value used for ROWS parameter changed from 64 to 12

Table TRSYS.WDS_HR_TEMP:
  154803 Rows successfully loaded.
   0 Rows not loaded due to data errors.
   0 Rows not loaded because all WHEN clauses were failed.
   0 Rows not loaded because all fields were null.


Space allocated for bind array:                 253872 bytes(12 rows)
Read   buffer bytes: 1048576

Total logical records skipped:          1
Total logical records read:        154803
Total logical records rejected:         0
Total logical records discarded:        0

Run began on Fri Dec 01 07:59:38 2017
Run ended on Fri Dec 01 08:00:07 2017

Elapsed time was:      00:00:28.45
CPU time was:          00:00:04.69
-bash-4.1$
```

"Completeness" TS-B.03 - Log of the uploading = Total records uploaded to the ORD staging table = 154,803 + 1 header record

Toad for Oracle

CECO@TRSYP.AIG.COM (trsyp1:liv1exa03) - Editor (New 1*)

CECO@PLUTUSP.AIG.COM (plutusp2:wlt5exa02) - Editor (New 1*)

New 1 *

```
select count(*) from olf_ext.ol_aig_wds_hr_temp
```

"Completeness" TS-B.04 - Count of records in OL staging table = 154,803

Data Grid

Data Grid | Query Viewer | Explain Plan | Script Output | Messages

COUNT(*)
154803

AutoCommit is OFF   CAPS NUM INS

CA-1742798 WDS to OL-S01.34

Desktop: SQL   CECO

New 1 *   New 2 *   +

```
1   •    select * from olf_ext.ol_aig_wds_hr_temp where wkr_ent_id in ('5225319','5254584','5256738','1401539')
2
3   ▶    select count(*) from olf_ext.ol_aig_gdw_hr h
4        where exists (select 1 from olf_ext.ol_aig_wds_hr_temp where wkr_ent_id = h.employee_id)
```

Data Grid

Data Grid   Query Viewer   Explain Plan   Script Output   Messages

COUNT(*)

▶   154803

"Completeness" TS-B.05 - Count of records in OL HR
table = 154,803

5:12 PM
7:22 PM
46:28 PM
9:34 PM
1:10 AM
4:38 AM
:06 PM
0:43 AM
0:57 AM
1:11 AM
1:26 AM
8:49 PM
9:21 PM
9:46 PM
:06 PM
6:49 PM
0:54 PM
:33 PM
2:17 PM

Commit is OFF   CAPS NUM INS

```
-rwxrwxrwx. 1 tmsprd    tms      71355583 Nov 30 07:59 V_WDS_CURRENT_WJG_v1.0.dat20171130
-rwxrwxrwx. 1 tmsprd    tms      71387115 Dec  1 07:59 V_WDS_CURRENT_WJG_v1.0.dat20171201
-bash-4.1$ cat V_WDS_CURRENT_WJG_v1.0.dat20171201|grep 5225319



-bash-4.1$ cat V_WDS_CURRENT_WJG_v1.0.dat20171201|grep 5254584
```

"Accuracy" TS-C.01 - WDS fields (1) WKR_ENT_ID = 1401539 (2) AD_Domain = R1-CORE (3) Work-office = 1000186 (4) Hire_Date = 2012-11-19 (5) Job_CP = FAGA0044 (6) Last_Update = 2017-12-01 07:15:41 (7) GEMS_ID = 23455 (8) Terminate_Date = 2017-11-29 (9) Last_Worked_Date = 2017-11-29 (10) Manager_ID = 5033824

```
-bash-4.1$ cat V_WDS_CURRENT_WJG_v1.0.dat20171201|grep 1401539
1401539|                        ||R1-CORE|                    ||1000186|2012-11-19|E|||||||272
                                        ||5033824|  ||2017-11-29|T|
29|2017-12-01 07:15:41||                                       |FAGA00
44|                                             |23455||
                                                        2012-11-19|T|29.759432|-95
.394609||||||
-bash-4.1$ cat V_WDS_CURRENT_WJG_v1.0.dat20171201|grep 5256738



61||||||
-bash-4.1$
```

CA-1742798 WDS to OL-S01.36

Desktop: SQL   CECO

New 1 *   New 2 *   +

```
1   select * from olf_ext.ol_aig_wds_hr_temp where wkr_ent_id in ('5225319','5254584','5256738','1401539')
2
3   select count(*) from olf_ext.ol_aig_gdw_hr h
4   where exists (select 1 from olf_ext.ol_aig_wds_hr_temp where wkr_ent_id = h.employee_id)
```

"Accuracy" TS-C.02 Query from staging table OLF_EXT.OL.AIG_WDS_HR_TEMP

Data Grid

Data Grid   Query Viewer   Explain Plan   Script Output   Messages

"Accuracy" TS-C.02 Results Query from the staging table OLF_EXT.OL.AIG_WDS_HR_TEMP - Full dump is in Pg. 24

WKR_ENT_ID	WKR_LAST_NM	WKR_FIRST_NM	WKR_MID_NM	AD_DOMAIN_TX	WKR_LAN_ID	WKR_EMAIL_TX	AIG_FAX_NO	AIG_MOBILE_NO	CBD	COST
5254584										
1401539				R1-CORE				088001024300		
5256738										
5225319										

5:12 PM
7:22 PM
46:28 PM
9:34 PM
1:10 AM
4:38 AM
:06 PM
0:43 AM
0:57 AM
1:11 AM
1:26 AM
8:49 PM
9:21 PM
9:46 PM
:06 PM
6:49 PM
0:54 PM
:33 PM
2:17 PM

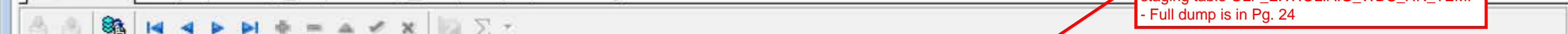Commit is OFF   CAPS   NUM   INS

"Accuracy" TS-C.03 Query from HR table OLF_EXT.OL_AIG_GDW_HR

"Accuracy" TS-C.03 Results Query from the HR table OLF_EXT.OL_AIG_GDW_HR- Full dump is in Pg. 40

```
select * from olf_ext.OL_AIG_GDW_HR where employee_id in ('5225319','5254584','5256738','1401539')
```

EMPLOYEE_ID	EMPLOYEE_TYPE	EMP_FIRST_NAME	EMP_LAST_NAME	EMP_MID_NAME	HIRE_DATE	JOB_CODE	JOB_TITLE	GLOBAL_JOB_GRADE	WWCC	GEMS_ID	BUSINESS_UNI
1401539	E				11/19/2012	FAGA0044			0880	23455	

CECO@PLUTUSP.AIG.COM (plutusp2:wlt5exa02) - Editor (New 1*)

Desktop: SQL   CECO

New 1 *   **New 2 ***

```
1   •   select * from olf.personnel
2       where last_update > to_date('12/1/2017','mm/dd/yyyy')
3
4       select * from olf.personnel_history where id_number in (21940,23271,23337,24155)
5       order by id_number, last_update
```

Data Grid

Data Grid | Query Viewer | Explain Plan | Script Output | Messages

"Accuracy" TS-C.04 Query the table "olf.Personnel_history" to obtain the update history of the employee record. The data dump is in Pg. 41

ID_NUMBER	NAME	TITLE	FIRST_NAME	LAST_NAME	PHONE	FAX	PERSONNEL_TYPE	STATUS	PASSWORD_DATE	AUTHORISER	LAST_UPDATE
21940											
21940											
23271											
23271											
23271											
23271											
23271											
23337							1	1	2/24/2016 10:46:57 AM	20020	2/24/2016 10:46:57
23337							1	1	2/24/2016 10:46:57 AM	20001	2/25/2016 8:15:27 AI
23337							1	1	2/24/2016 10:46:57 AM	20004	12/1/2017 8:15:19 AI

oCommit is OFF   CAPS NUM INS

WKR_ENT_ID	AD_DOMAIN_TX	COST_CENTER_CD	WKR_OFFC_CD	HIRE_DT	WKR_TYPE_CD	FULL_PART_TIME_IN	MANAGER_ENT_ID	TERMINATION_DT	WKR_STATUS_CD	CREATE_TS	LAST_WORKED_DT	LAST_UPDATE_TS	JOB_CD	JOB_GRADE_NO	WORLD_WIDE_COMPANY_CD	GEMS_ID
5254584	R3-CORE	10027057	1001845	12/08/2015	E	F	5256585		A	2015-11-27 02:52:44		2017-12-01 07:15:41	FAGA0044	19	3741	48199
1401539	R1-CORE		1000186	11/19/2012	E	F	5033824	11/29/2017	T	2015-10-04 11:07:54	11/29/2017	2017-12-01 07:15:41	FAGA0044	19	0880	23455
5256738	R3-CORE	7075	1001845	02/16/2016	E	F	5256678		A	2016-01-19 02:51:47		2017-12-01 07:15:41	FAGA0043	18	3741	48199
5225319	R1-CORE	10027063	1002549	06/01/2014	E	F	5238039		A	2015-10-04 11:09:11		2017-12-01 07:15:18	RERN4304	19	3741	48199

"Accuracy" TS-C.02 Results Query from staging table OLF_EXT.OL.AIG_WDS_HR_TEMP (shown in Pg. 36)

[1] EMPLOYEE_ID	EMPLOYEE_TYPE	[3] HIRE_DATE	[4] JOB_CODE	WWCC	[7] GEMS_ID	BUSINESS_UNIT	REGION_CODE	SHELL_CODE	EMP_STATUS	[9] LAST_DAY_WORK	LAST_DAY_PAID	[8] TERMINATION_DATE	PART_FULL_INDICATOR	[10] MANAGER_ID
1401539	E	11/19/2012	FAGA0044	0880	23455	Finance		31100000	T	11/29/2017		11/29/2017	F	5033824
5225319	E	06/01/2014	RERN4304	3741	48199	Finance	42	03425510	A				F	5238039
5254584	E	12/08/2015	FAGA0044	3741	48199	Finance	42	03425510	A				F	5256585
5256738	E	02/16/2016	FAGA0043	3741	48199	Finance	42	03425510	A				F	5256678

"Completeness" TS-B.06 - Of the 4 records with changes from 2017-12-01 WDS, only 1 record has status = "Terminated"

"Accuracy" TS-C.03 Results Query from staging table OLF_EXT.OL.AIG_GDW_HR (shown in Pg. 37)

ID_NUM BER	PERS ONNE L TYP	STATU S	PASSWORD_ DATE	AUTHORIS ER	LAST_UPD ATE	PERSO NNEL_ VERSIO	EMPLOY EE_ID	STATE_ID	COUNTRY	MAIL_COD E	TSV_GROU P_ID	PRIORITY_ USER	PASSWOR D_NEVER_ EXPIRES	TIME_ZONE	PHOTO_NO DE_ID	SIGNATUR E_NODE_ID	CITIZENSHI P_COUNTR Y	RESIDENC E_COUNTR Y
21940	1	1	10/09/2014	20,020	10/09/2014	1	5225319	-1	-1		-1	-1	-1	0	0	0	0	0
21940	1	1	10/09/2014	20,004	09/12/2015	2	5225319	-1	-1		-1	-1	-1	0	0	0	0	0
21940	1	1	10/09/2014	20,001	02/16/2016	3	5225319	-1	-1		-1	-1	-1	0	0	0	0	0
21940	1	1	10/09/2014	20,004	12/01/2017	4	5225319	-1	-1		-1	-1	-1	0	0	0	0	0
23271	0	1	01/12/2016	20,020	01/12/2016	1	5254584	0	0		0	0	0	0	0	0	0	0
23271	0	1	01/12/2016	20,004	01/13/2016	2	5254584	0	0		0	0	0	0	0	0	0	0
23271	1	1	01/12/2016	20,020	02/24/2016	3	5254584	0	0		0	0	0	0	0	0	0	0
23271	1	1	01/12/2016	20,004	11/30/2017	4	5254584	0	0		0	0	0	0	0	0	0	0
23271	1	1	01/12/2016	20,004	12/01/2017	5	5254584	0	0		0	0	0	0	0	0	0	0
23337	1	1	02/24/2016	20,020	02/24/2016	1	5256738	0	0		0	0	0	0	0	0	0	0
23337	1	1	02/24/2016	20,001	02/25/2016	2	5256738	0	0		0	0	0	0	0	0	0	0
23337	1	1	02/24/2016	20,004	12/01/2017	3	5256738	0	0		0	0	0	0	0	0	0	0
24155	1	1	11/10/2017	23,925	11/10/2017	1	1401539	0	0		0	0	0	0	0	0	0	0
24155	1	1	11/10/2017	20,004	11/11/2017	2	1401539	0	0		0	0	0	0	0	0	0	0
24155	1	2	11/10/2017	20,004	12/01/2017	3	1401539	0	0		0	0	0	0	0	0	0	0

"Accuracy" TS-C.04 Results from query
OLF_PERSONNEL_HISTORY (pg. 38)

Personnel Editor -

File   View   Help

Type: Internal   License Type: Full Access   ☐ Priority User

Personnel Id: 24155   Version: 3

Short Ref Name:

Short Alias Name:

Status: Do not use

Employee Id: 1401539

Title: Accountant III

Last Name:

Forename:

Citizenship:

Residency:

Tax Filing Countries:

X_____

☐ Password
  Never Expires

Address:

City:

State: None

Country:

Mail Code:

Email:

Phone:

Fax:

Time Zone: NONE

TSV Peer Group: None

| Business Units | Legal Entities | Portfolios | Security Groups | Secured Index | Functional Groups | User Defaults | Personnel Info | Trading |

Available Business Units	Selected Business Units
AAI.AU.2- BU	AIG.CSMC20.US.2- BU
AGLICDELAW.US.2- BU	AIG.HAMILT.US.1- BU
AIG.271OFF.US.1- BU	AIG.HAMILT.US.2- BU
AIG.AGCORP.US.1- BU	AIG.HAMILT.US.3- BU
AIG.AGEMPL.US.1- BU	AIG.HAMILT.US.4- BU
AIG.AGINVE.US.10- BU	HAMILTONIN.US.2- BU
AIG.AGREAL.US.1- BU	PC.GREMXIP.US.2- BU
AIG.AICCO.US.2- BU	PC.GREMXIP.US.3- BU

Add >>

Remove <<

ⓘ Modified by 'FINPRD_SRV3' at 12/1/2017 8:15:23 AM

Personnel Editor

File   View   Help

Type: Internal   **License Type:** Full Access   ☐ Priority User

Personnel Id: 24155   **Version:** 3

Short Ref Name:

Short Alias Name:

Status: Do not use

Employee Id: 1401539

Title: Accountant III

Last Name:

Forename:

Citizenship:

Residency:

Tax Filing Countries:

x_____

☐ Password Never Expires

Address:

City:

State: None

Country:

Mail Code:

Email:

Phone:

Fax:

Time Zone: NONE

TSV Peer Group: None

| Business Units | Legal Entities | Portfolios | Security Groups | Secured Index | Functional Groups | User Defaults | Personnel Info | Trading |

Info Type △	Value	
Termination Date	11/29/2017	
Wire Interface ID		

File   Edit   Search   Editor   Session   Database   Debug   View   Utilities
Rerun   Window   Help

CA-1742798 WDS to OL-S01.44
<No Workspace selected>

CECO@PLUTUSP.AIG.COM (plutusp1:wlt5exa01) - Editor (New 4 *)

Desktop: SQL

CECO

New 1 *   New 2 *   New 3 *   New 4 *

```sql
1  select    a.employee_id as approver_code
2          , upper(a.title) as approver_title
3          , upper(a.last_name) || ', ' || upper(a.first_name) as approver_name
4          , case when (d.name = 'Authorized') then 'Y' else 'N' end as active
5          , c.emp_status,a.last_update
6    from      olf.personnel a, olf_ext.ol_aig_gdw_hr c, olf.personnel_status d
7    where     a.employee_id = c.employee_id
8          and a.employee_id='1401539'
9          and a.status = d.id_number
10   and   exists ( select * from olf.account_signatory b
11               where a.id_number = b.account_signatory )
```

Data Grid

Data Grid   Query Viewer   Explain Plan   Script Output   Messages

APPROVER_CODE	APPROVER_TITLE	APPROVER_NAME	ACTIVE	EMP_STATUS	LAST_UPDATE

"Completeness" TS-B.06 - The "Terminated" staff from the 2017-12-01 WDS file is not a signatory in OL

AutoCommit is OFF   CAPS   NUM   INS