Oracle FLEXCUBE Investor Servicing® Upload Adapter Development Guide

Release 12.0

April 2012

Oracle Part Number E51528-01

Contents

1 Preface 3
  1.1 Audience 3
  1.2 Related documents 3
  1.3 Conventions 3
  1.4 Hypothetical Example used 3
2 Introduction 4
  2.1 How to use this Guide 4
3 Upload Adapter Overview 4
  3.1 Creation of Data Source 5
  3.2 Creating the Excel sheet 7
  3.3 Hand coding the package 9
    3.3.1 ***STATIC DATA*** *9*
    3.3.2 ***Edit the SPFMGBULKIMPORTWRAPPERfunction*** *9*
  3.4 Upload from Excel to Upload Table 11
  3.5 CVDUPLOD through Flexcube screen 13
4 Back end Flow 14
5 Check list 14

1 Preface

This document upload adapter development Guide explains the steps to create spreadsheet based upload macro that helps to upload data into FLEXCUBE IS. These uploads are used in migrations.

1.1 Audience

This upload adapter development is intended for FLEXCUBE Application Developers/Users who require to do the following tasks:

- Create spreadsheet that uploads data into FLEXCUBE IS for identified tables

1.2 Related documents

For more information on Interfaces, see these resources:

- *FCIS-FD01-01-01-Development Overview Guide*
- *FCIS-FD04-01-01-Interface Getting started*

1.3 Conventions

The following text conventions are used in this document:

Convention	**Meaning**
boldface	Boldface type indicates graphical user interface elements (for example, menus and menu items, buttons, tabs, dialog controls), including options that you select.
italic	italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates language and syntax elements, directory and file names, URLs, text that appears on the screen, or text that you enter.

1.4 Hypothetical Example used

The following is the work example used in this document.

- FLEXCUBE IS has the function ID UTDATREP (Authorized Representative Details).
  This is a maintenance function ID. This requires upload adapter spreadsheet.

Spread sheet with work sheet 'BULKAUTHREPIMPORTTBL'is created.

2 Introduction

2.1 How to use this Guide

- Chapter 2, "Introduction"

  This is an introduction section.
- Chapter 3, "Upload adapter Overview"

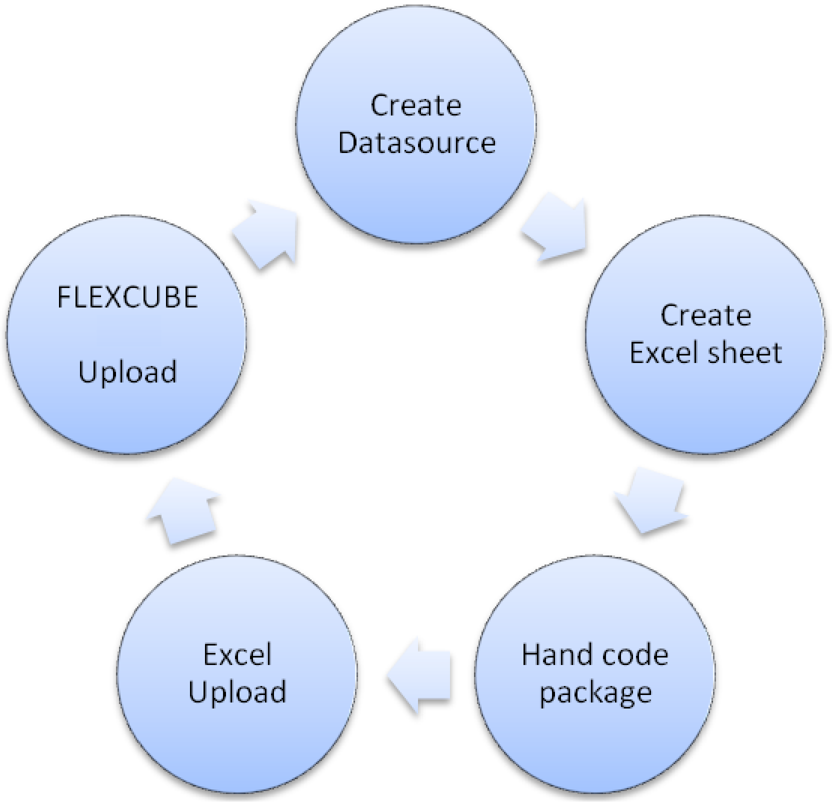
  This section describes the Upload adapter
- Chapter 4, "Back end Flow"

  This section describes the backend flow.
- Chapter 4, "Check List"

  This section contains the checklist for upload adapter

3 Upload Adapter Overview

This section describes the overview of the Upload adapter

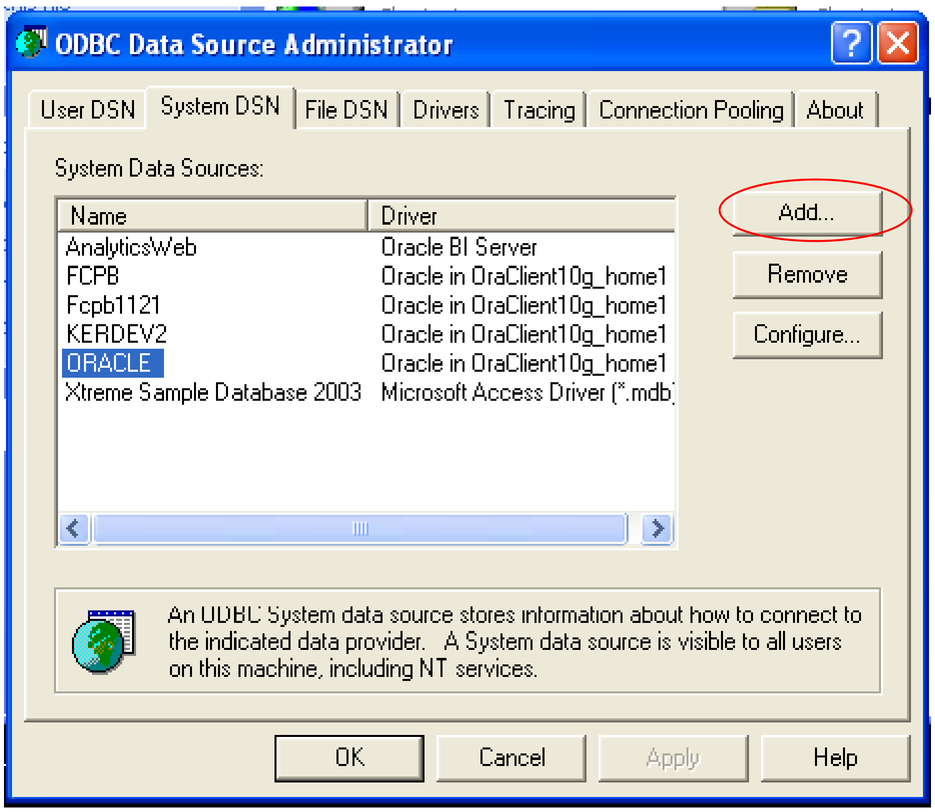


3.1 Creation of Data Source

Create an ODBC Data Source

An ODBC data source is needed to import schema information about a data source into an Oracle Database. To create an ODBC data source, perform the following steps:

- Click **Start > Control Panel > Administrative Tools > Data Sources (ODBC)** to open the ODBC Data Source Administrator



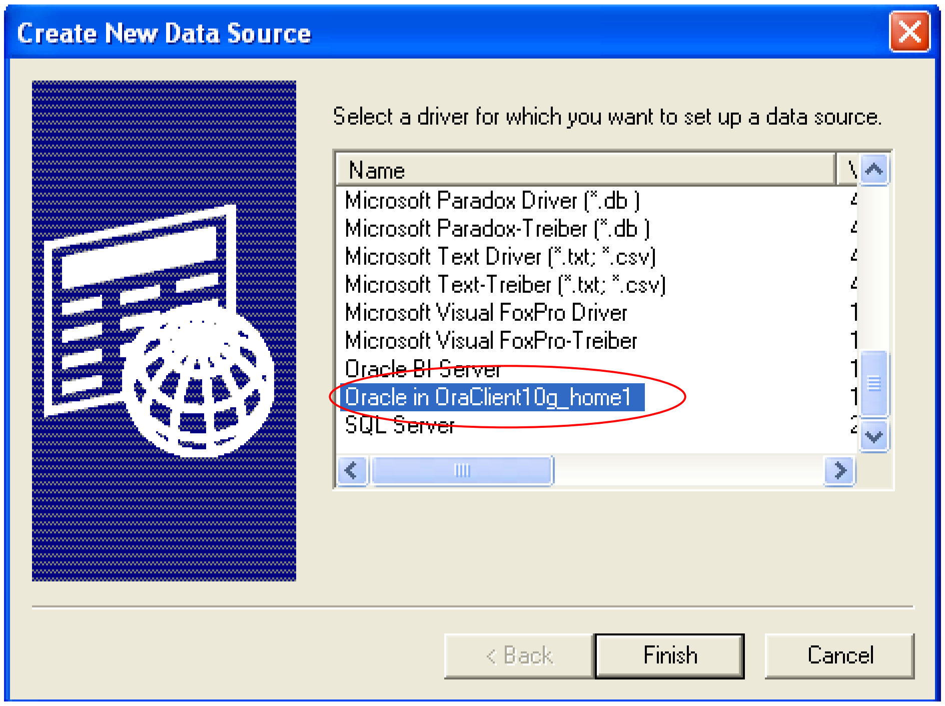
Name	Driver
AnalyticsWeb	Oracle BI Server
FCPB	Oracle in OraClient10g_home1
Fcpb1121	Oracle in OraClient10g_home1
KERDEV2	Oracle in OraClient10g_home1
ORACLE	Oracle in OraClient10g_home1
Xtreme Sample Database 2003	Microsoft Access Driver (*.mdb)

An ODBC System data source stores information about how to connect to the indicated data provider. A System data source is visible to all users on this machine, including NT services.

The following screen appears.
We need to add a new Data Source Name

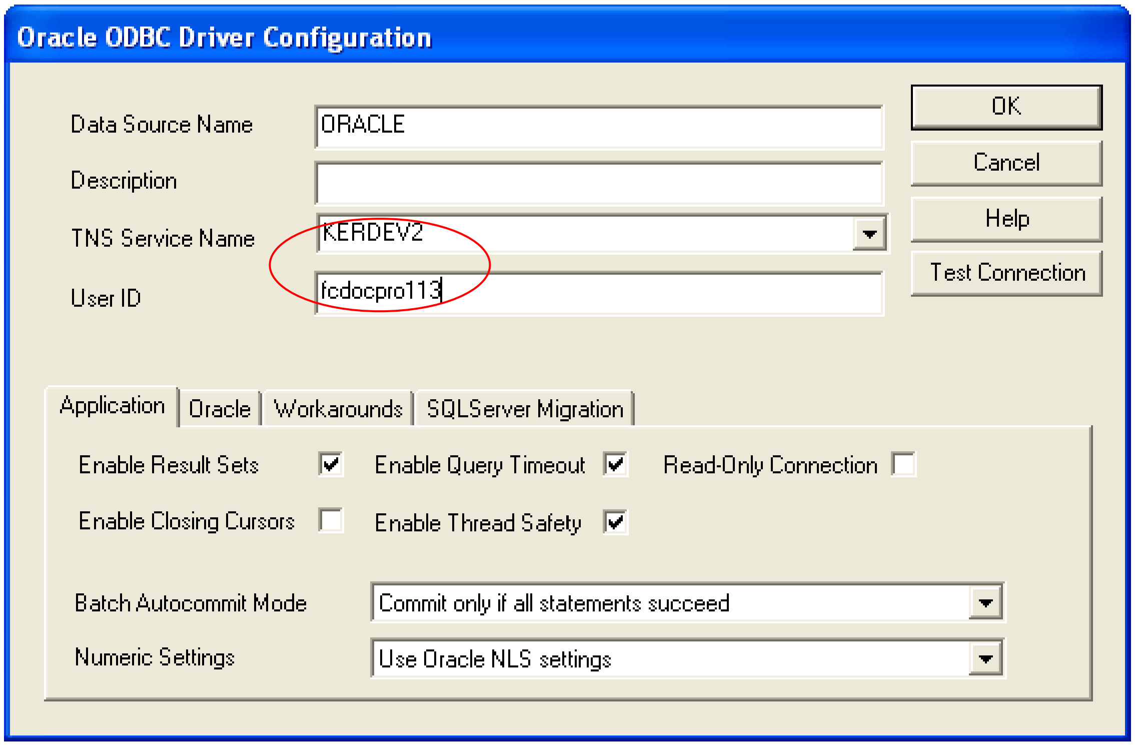
- Click the **System DSN** tab and click **Add**.

The Create New Data Source dialog box opens.


Create New Data Source

Select a driver for which you want to set up a data source.

Name
Microsoft Paradox Driver (*.db)
Microsoft Paradox-Treiber (*.db)
Microsoft Text Driver (*.txt; *.csv)
Microsoft Text-Treiber (*.txt; *.csv)
Microsoft Visual FoxPro Driver
Microsoft Visual FoxPro-Treiber
Oracle BI Server
Oracle in OraClient10g_home1
SQL Server


Oracle ODBC Driver Configuration

Data Source Name	ORACLE
Description	
TNS Service Name	KERDEV2
User ID	fcdocpro113

Enable Result Sets ✓ Enable Query Timeout ✓ Read-Only Connection ☐

Enable Closing Cursors ☐ Enable Thread Safety ✓

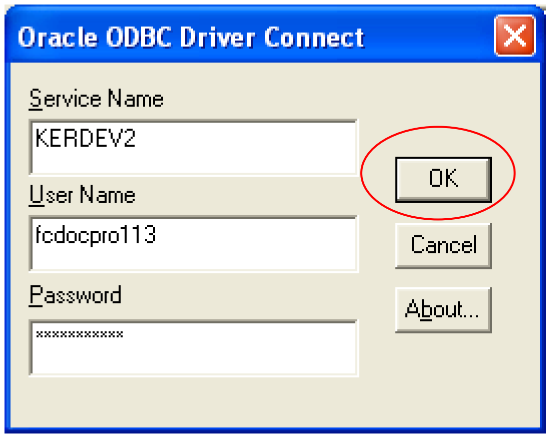
Batch Autocommit Mode: Commit only if all statements succeed

Numeric Settings: Use Oracle NLS settings

Data Source Name: Name for the DSN
TNS service Name: The TNS name currently working on, where the table exists.
User ID: Schema name for the TNS

On adding it Check for the Connection.

Click on Test Connection

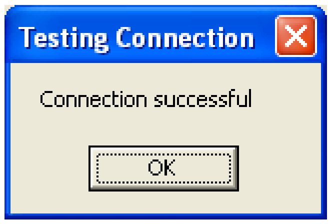

Oracle ODBC Driver Connect

Service Name: KERDEV2

User Name: fcdocpro113

Password: ***********

Enter the user name and password for the schema and Click OK


Testing Connection

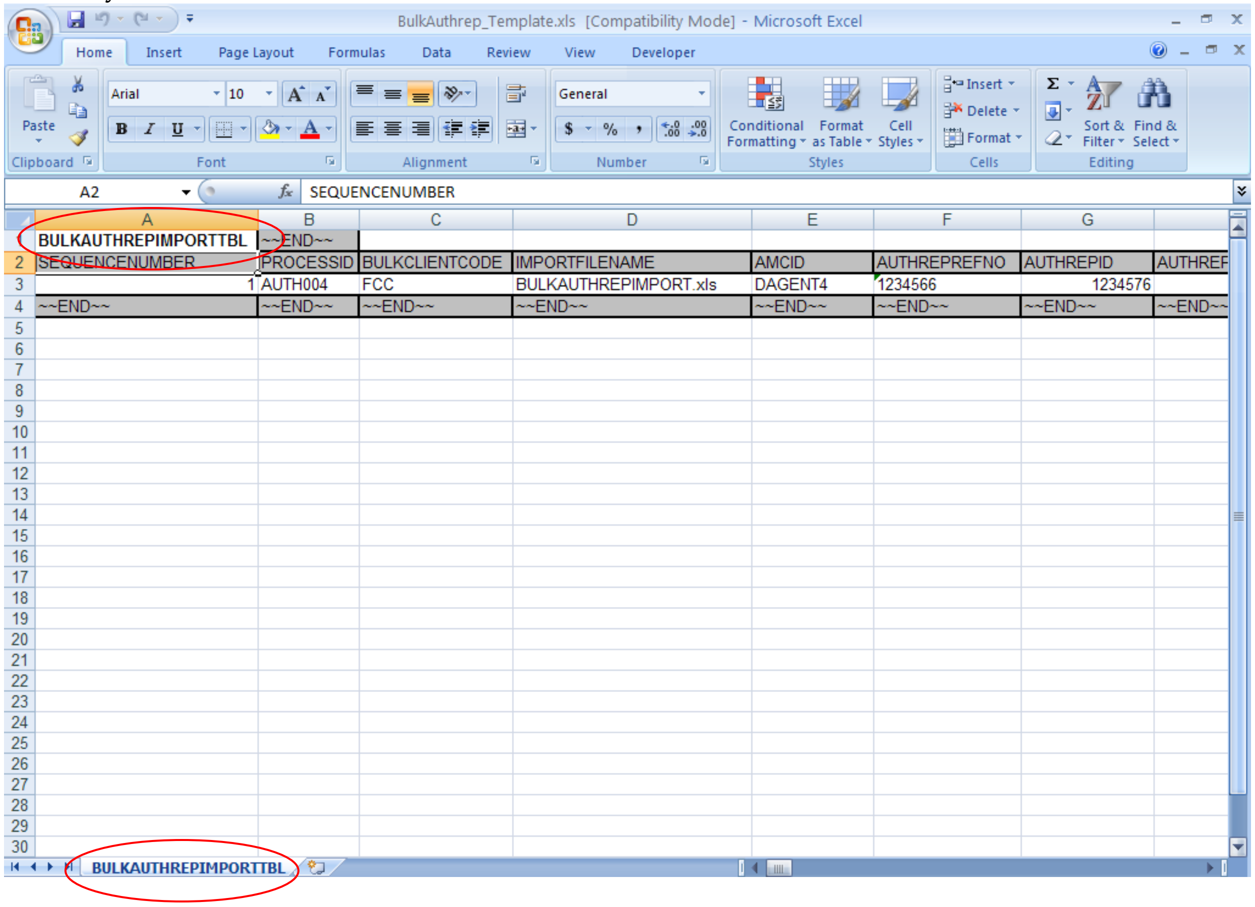
Connection successful

This message is displayed on successful connection.

3.2 Creating the Excel sheet

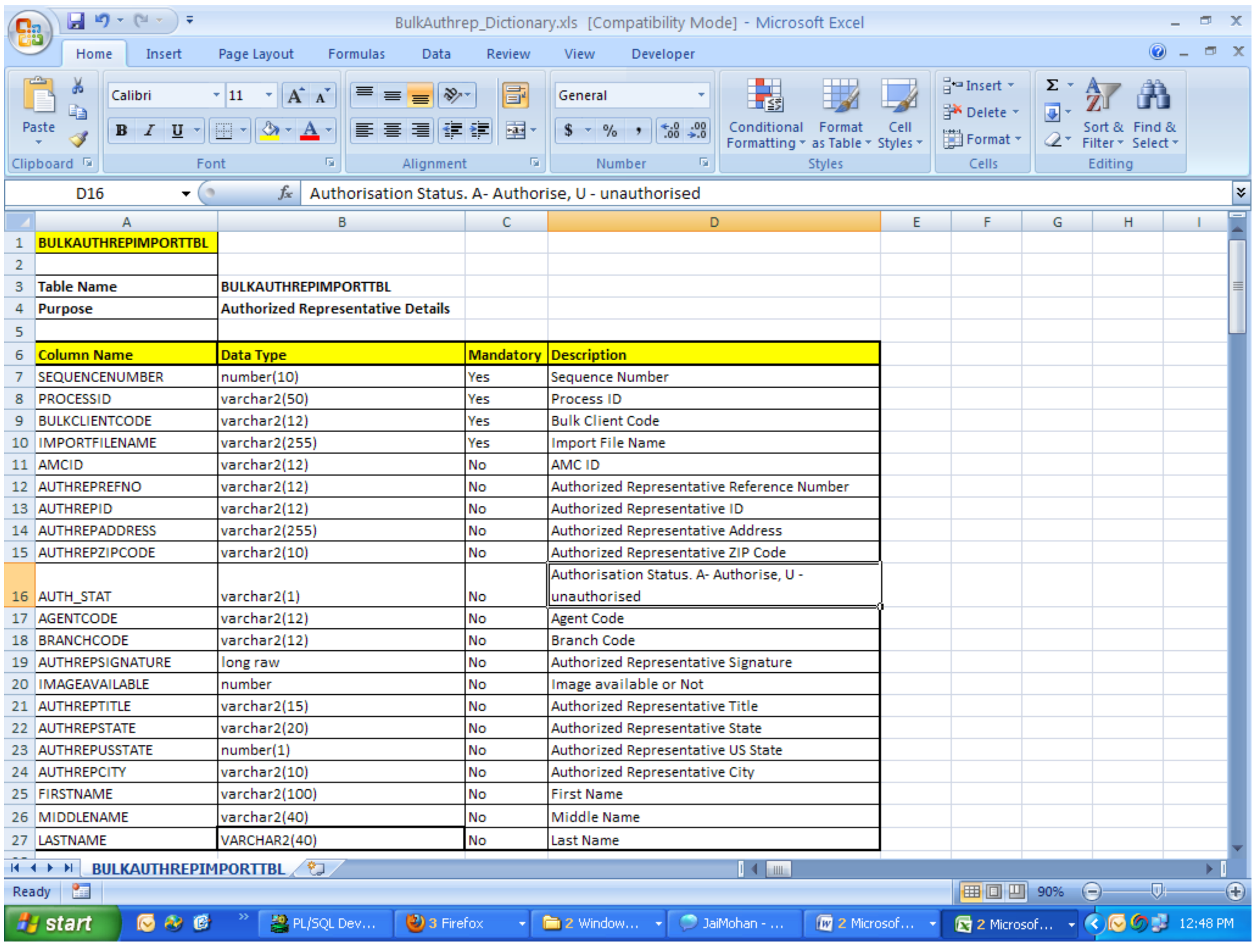
Prepare the EXCEL sheet as shown in the below format. With the sheet name as the table name.
Grid A1 contains the name of the table where the data is to be uploaded.
~~ marks the end of the columns.

The data from the sheet will be uploaded to the upload table. The processid should be unique for every record.



A	B	C	D	E	F	G	H
BULKAUTHREPIMPORTTBL	~~END~~						
SEQUENCENUMBER	PROCESSID	BULKCLIENTCODE	IMPORTFILENAME	AMCID	AUTHREPREFNO	AUTHREPID	AUTHREF
1	AUTH004	FCC	BULKAUTHREPIMPORT.xls	DAGENT4	1234566	1234576	
~~END~~	~~END~~	~~END~~	~~END~~	~~END~~	~~END~~	~~END~~	~~END~~

The dictionary excel sheet contains the details of the upload table;



D16 — Authorisation Status. A- Authorise, U - unauthorised

BULKAUTHREPIMPORTTBL

Table Name	BULKAUTHREPIMPORTTBL
Purpose	Authorized Representative Details

Column Name	Data Type	Mandatory	Description
SEQUENCENUMBER	number(10)	Yes	Sequence Number
PROCESSID	varchar2(50)	Yes	Process ID
BULKCLIENTCODE	varchar2(12)	Yes	Bulk Client Code
IMPORTFILENAME	varchar2(255)	Yes	Import File Name
AMCID	varchar2(12)	No	AMC ID
AUTHREPREFNO	varchar2(12)	No	Authorized Representative Reference Number
AUTHREPID	varchar2(12)	No	Authorized Representative ID
AUTHREPADDRESS	varchar2(255)	No	Authorized Representative Address
AUTHREPZIPCODE	varchar2(10)	No	Authorized Representative ZIP Code
AUTH_STAT	varchar2(1)	No	Authorisation Status. A- Authorise, U - unauthorised
AGENTCODE	varchar2(12)	No	Agent Code
BRANCHCODE	varchar2(12)	No	Branch Code
AUTHREPSIGNATURE	long raw	No	Authorized Representative Signature
IMAGEAVAILABLE	number	No	Image available or Not
AUTHREPTITLE	varchar2(15)	No	Authorized Representative Title
AUTHREPSTATE	varchar2(20)	No	Authorized Representative State
AUTHREPUSSTATE	number(1)	No	Authorized Representative US State
AUTHREPCITY	varchar2(10)	No	Authorized Representative City
FIRSTNAME	varchar2(100)	No	First Name
MIDDLENAME	varchar2(40)	No	Middle Name
LASTNAME	VARCHAR2(40)	No	Last Name

3.3 Hand coding the package

An upload package is hand coded for insertion of data from the upload table to the master table.

3.3.1 STATIC DATA

A Static data should be inserted into the Paramstbl with the paramcode – UPLOADFORMAT for the respective upload with the unique Paramvalue.

Paramcode	Paramvalue	Paramtext	Paramlanguage	Sortorder
UPLOADFORMAT	AUTHREP	Authorized Representative	1033	1

3.3.2 Edit the SPFMGBULKIMPORTWRAPPER function

SPFMGBULKIMPORTWRAPPER function needs to be edited so that the upload for the user created function id can be constructed.

The functions to be changed :

- SFLOGPROCESSIDS
- SFPROCESS

<u>SFLOGPROCESSIDS</u>

e.g.
Create a cursor to pick the value from the table BULKAUTHREPIMPORTTBL :

```
    SELECT PROCESSID, BULKCLIENTCODE, IMPORTFILENAME, COUNT(*) TOTALRECS
        FROM BULKAUTHREPIMPORTTBL A
      WHERE NOT EXISTS (SELECT 1
                FROM BULKIMPORTLOGTBL B
              WHERE B.PROCESSID = A.PROCESSID)
      GROUP BY PROCESSID, BULKCLIENTCODE, IMPORTFILENAME;
```

Then ,

```
  ELSIF IPLTYPEOFDATA = 'AUTHREP' THEN
      DBG('In type of data AUTHREP');

      FOR EACHPROCESS IN C_AUTHREPDATA LOOP
        DBG('EachProcess.ProcessID  ' || EACHPROCESS.PROCESSID);
        DBG('v_ProcessID  ' || V_PROCESSID);
        IF (V_PROCESSID = EACHPROCESS.PROCESSID) THEN
          ROLLBACK TO LOGPROCESS;
          OPERRORCODE := 'E-DUPLICATEBULK';
          OPERRORMSG  := 'Duplicate data for BulkClientCode/ImportFileName';
          RETURN FALSE;
        ELSE
          V_PROCESSID := EACHPROCESS.PROCESSID;
        END IF;

        IF IPDUPCHECK = 'Y' THEN
          IF NOT SFFILEDUPCHECK(EACHPROCESS.IMPORTFILENAME, IPLTYPEOFDATA)
THEN
            DBG('After return false from sfFileDupCheck');
            ROLLBACK TO LOGPROCESS;
            RETURN FALSE;
          END IF;
        END IF;

        IF NOT SFBULKIMPORTLOG(EACHPROCESS.PROCESSID,
                               EACHPROCESS.IMPORTFILENAME,
                               EACHPROCESS.BULKCLIENTCODE,
                               EACHPROCESS.TOTALRECS,
                               IPLTYPEOFDATA) THEN
```

```
                    ROLLBACK TO LOGPROCESS;
                    RETURN FALSE;
                END IF;
            END LOOP;
END IF;
```

SFPROCESS

Call the respective package using the below code

```
    ELSIF IPLTYPEOFDATA = 'AUTHREP' THEN
                        DBG('Calling PkgAuthorizedRepresentative.spBulkImportAuthRep');

    PKGAUTHORIZEDREPRESENTATIVE.SPBULKIMPORTAUTHREP(EACHPROCESS.PROCESSID,

    IPKEYSTRING,
    IPMODULEID,
   IPCOMMITFLAG);
```

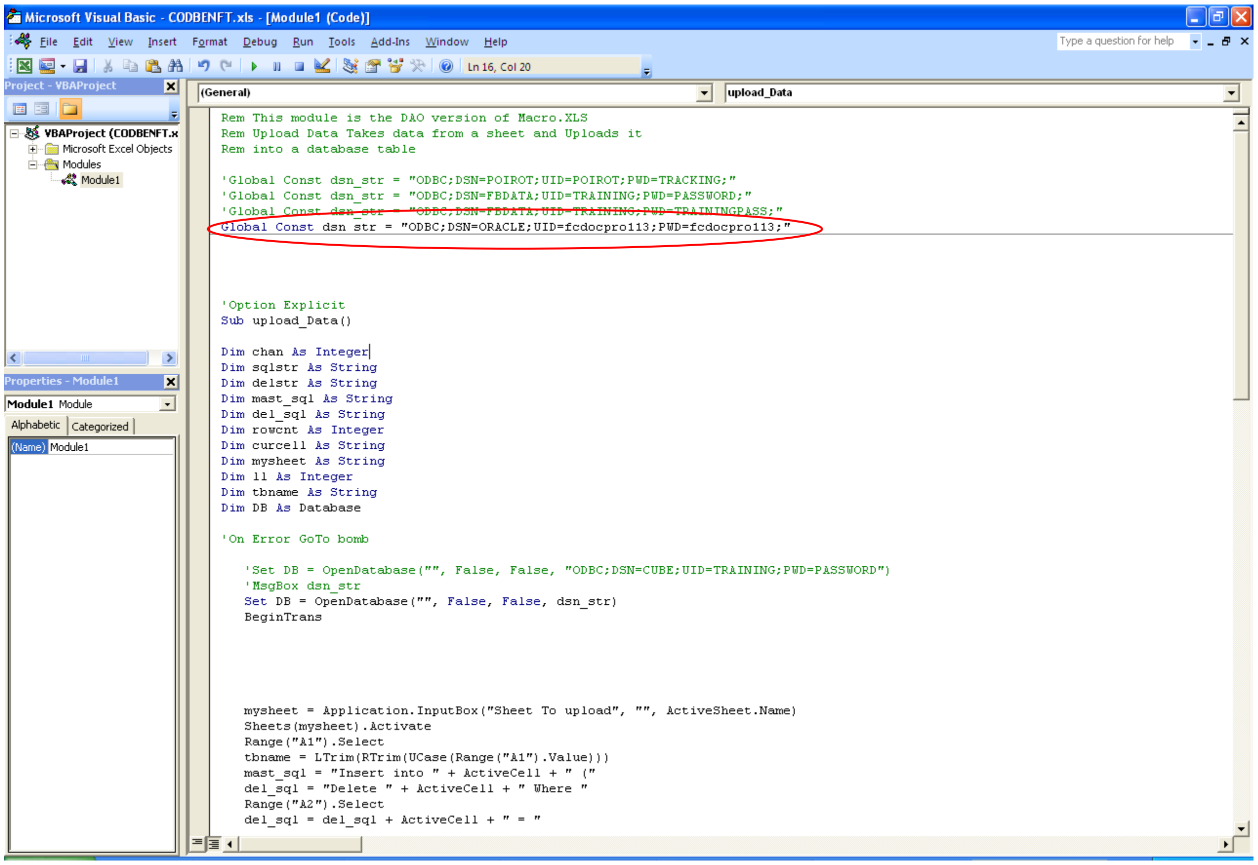
The user created function id is enabled for upload through the hand coded package.
Here the function id of the user is added for upload construction such that the function id and action code is passed to the hand coded package for upload.

3.4 Upload from Excel to Upload Table

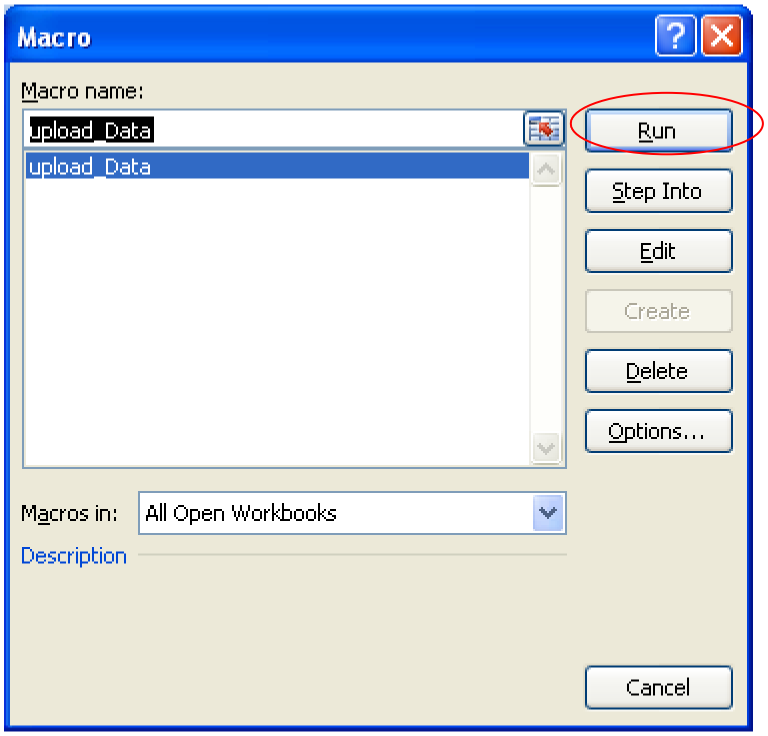
- Click on Developer → Visual basic

  Enable the macros in excel sheet.

  Create a module with macro name as Upload_data

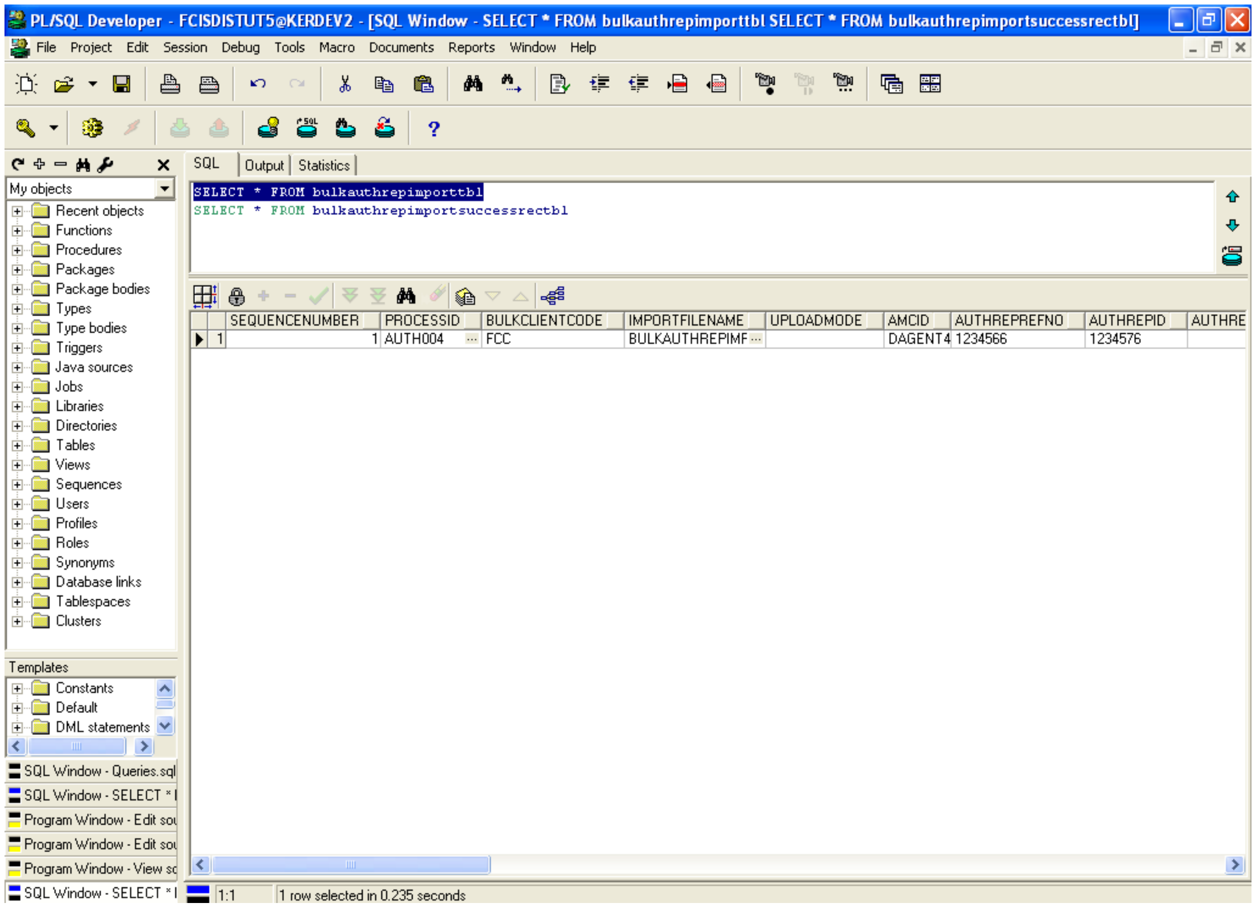


```
Rem This module is the DAO version of Macro.XLS
Rem Upload Data Takes data from a sheet and Uploads it
Rem into a database table

'Global Const dsn_str = "ODBC;DSN=POIROT;UID=POIROT;PWD=TRACKING;"
'Global Const dsn_str = "ODBC;DSN=FBDATA;UID=TRAINING;PWD=PASSWORD;"
'Global Const dsn_str = "ODBC;DSN=FBDATA;UID=TRAINING;PWD=TRAININGPASS;"
Global Const dsn_str = "ODBC;DSN=ORACLE;UID=fcdocpro113;PWD=fcdocpro113;"



'Option Explicit
Sub upload_Data()

Dim chan As Integer
Dim sqlstr As String
Dim delstr As String
Dim mast_sql As String
Dim del_sql As String
Dim rowcnt As Integer
Dim curcell As String
Dim mysheet As String
Dim ll As Integer
Dim tbname As String
Dim DB As Database

'On Error GoTo bomb

   'Set DB = OpenDatabase("", False, False, "ODBC;DSN=CUBE;UID=TRAINING;PWD=PASSWORD")
   'MsgBox dsn_str
   Set DB = OpenDatabase("", False, False, dsn_str)
   BeginTrans





   mysheet = Application.InputBox("Sheet To upload", "", ActiveSheet.Name)
   Sheets(mysheet).Activate
   Range("A1").Select
   tbname = LTrim(RTrim(UCase(Range("A1").Value)))
   mast_sql = "Insert into " + ActiveCell + " ("
   del_sql = "Delete " + ActiveCell + " Where "
   Range("A2").Select
   del_sql = del_sql + ActiveCell + " = "
```



Run the macro created with the name Upload_data
Click on RUN

Specify the uploading sheet name from which data is to be inserted in the table.



PL/SQL Developer - FCISDISTUT5@KERDEV2 - [SQL Window - SELECT * FROM bulkauthrepimporttbl SELECT * FROM bulkauthrepimportsuccessrectbl]

```
SELECT * FROM bulkauthrepimporttbl
SELECT * FROM bulkauthrepimportsuccessrectbl
```

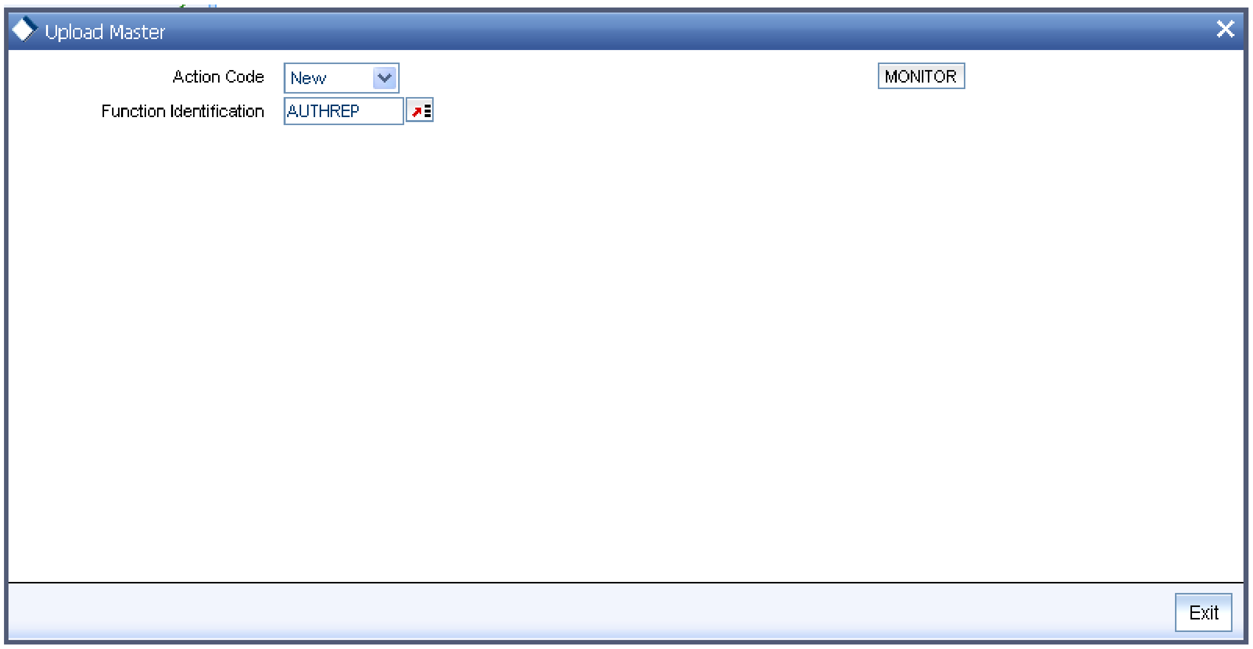
	SEQUENCENUMBER	PROCESSID	BULKCLIENTCODE	IMPORTFILENAME	UPLOADMODE	AMCID	AUTHREPREFNO	AUTHREPID	AUTHRE
1	1	AUTH004	FCC	BULKAUTHREPIMF		DAGENT4	1234566	1234576	

1:1 1 row selected in 0.235 seconds

Now check the table for successful insertion of data from the excel sheet.

3.5 CVDUPLOD through Flexcube screen

Data is uploaded to the master table through FLEXCUBE using the function id 'CVDUPLOD'



Action Code: New

Function Identification: AUTHREP

The function id is mentioned for which the table is to be uploaded.
The action code is New

Click on Save option in Flexcube.
After saving it will move the record into (respective success tables):

Select * from bulkauthrepimportsuccessrectbl
The table describes the details of the Upload made for the Function Id.

4 Back end Flow

cvpks_cvduplod_kernel => Calls the spfmgBulkimportwrapcall with the action code and function ID passed from the frontend.
spfmgBulkimportwrapcall => calls spFmgBulkImportWrapper
spFmgBulkImportWrapper => calls main package

cvpks_cvduplod_kernel.fn_main = > this package checks the function ID and action code
spFmgBulkImportWrapper. SFPROCESS = > calls
PKGAUTHORIZEDREPRESENTATIVE.SPBULKIMPORTAUTHRE which in turn calls
utpks_utdatrep_main.fn_main to upload the Authorized Representative details.

5 Check list

Specification	Data
Tables /Views created in Database? (Yes/No)	Yes

Tables /Views(which are to be uploaded) should be created in Database	
Data source available for Oracle database?(Yes/No) *ODBC data source is created for the uploading the excel sheet check for its availability.*	Yes
Creation of Excel Template ?(Yes/No) *Specify the name of the excel sheet and data format from which data is to be uploaded into the upload table.*	Yes
Enabling the macros and visual basic script? (Yes/No) *Script for connecting the excel sheet to the database schema for insertion of data*	Yes
Run the macro created *If successful it will upload the data from the sheet to the upload table else check the debug*	Macro name:Upload_data
Check the upload table? (Yes/No) *For successful update to the upload tables check the upload table*	Yes
Create a package for the function id? *Hand coded package should be created for the specific function id so as to move the data from the upload table to the base table.*	Yes Package name: pkgAuthorizedRepresentative
Edit the Cvpks_Cvduplod_Kernel package? (Yes/No) *Edit the package for function id through which upload has to be done in Flexcube environment.*	Yes
Upload through Flexcube? (Yes/No) *Launch the Flexcube screen for function id CVDUPLOD and specify the details for the function id and action*	**Yes**

Check for Successful upload?(Yes/No) *Check the respective success log table for the particular process ID.eg) bulkauthrepimportsuccessrectbl*	*Yes*
Check for successful upload?(Yes/No) *Check the table for which upload is done.*	Yes

FCIS-FD04-03-01-Upload Adapter Development Guide
April 2012
12.0

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
www.oracle.com/ financial_services/