

Loading Excel File into RED Warehouse

WhereScape RED can access and load into warehouse virtually any type of data. One of the most common sources that data loaded from is Excel file. WhereScape can quite gracefully handle almost all the cases of Excel data sources.

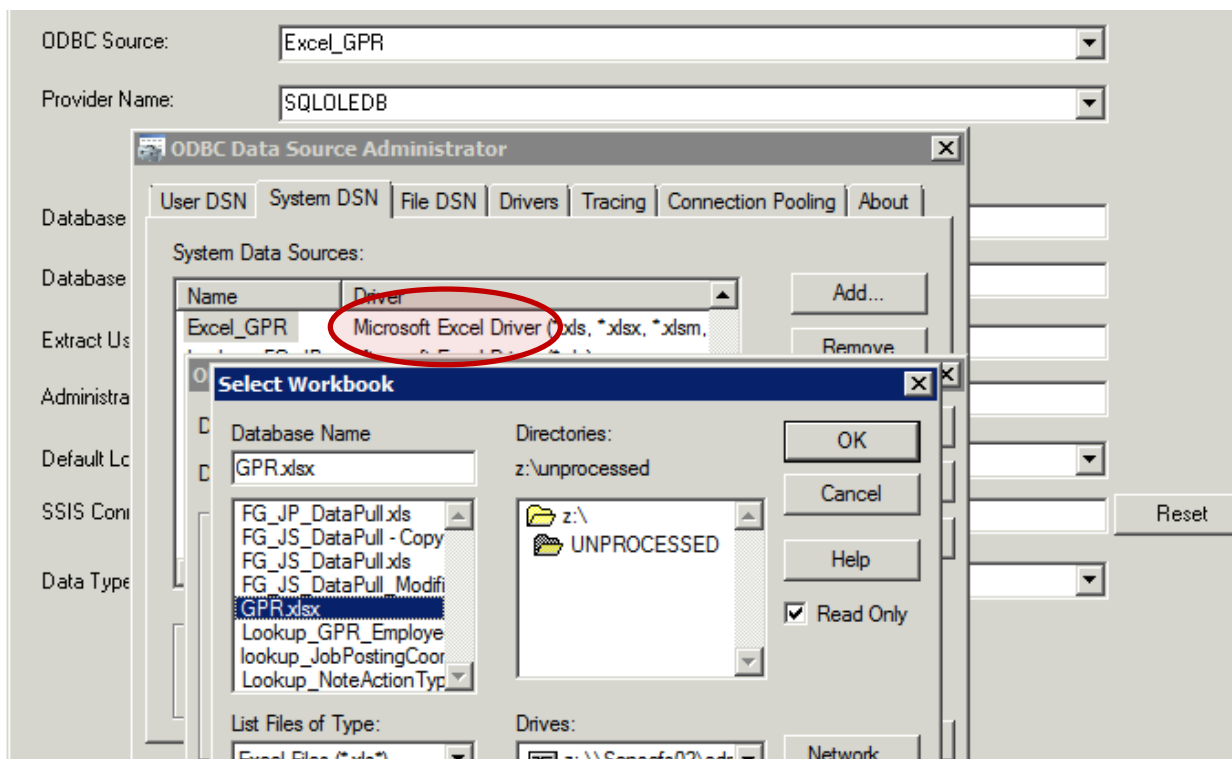
WhereScape has accumulated vast knowledge in dealing with multitude of data sources, including Excel files of all shapes and sizes. Most of this knowledge is embedded in the software itself, making it easy for our customers to get the needed data, its metadata and load it into WhereScape RED warehouse.

RED connections -- a first RED object created -- are used, among other purposes, to browse source data to obtain metadata and to load data into RED warehouse. WhereScape RED provides several types of connections to access Excel spreadsheets. While choosing the right method of loading Excel file, the consideration must be made based on volume and speed of loading, available connectivity options, ability to handle inconsistent data, error capturing, etc.

Below are few cases on how to configure loading of the Excel data into RED load_% tables.

1. ODBC-based Load

Consider Excel file -- a workbook gpr.xlsx -- that consists of multiple worksheets (Sheet1, Sheet2, and Sheet3). The first method we consider is ODBC-based Load. First what we need to do is to create ODBC connection using, for instance, Microsoft Excel Driver.



In the RED connection definitions choose “Native ODBC” option. When applicable, Native ODBC is a very fast option since it is using Bulk Insert in SQL Server, SQL*LOADER in Oracle or LOAD statement in DB2. But, because it is processing the whole intermediate file at once, the data has to be relatively “clean”: error in data cause the whole load to fail.

The dialog box 'Excel_GPR Definition' has a 'Connection Properties' tab. It contains the following fields:

- ☐ Data Warehouse
- Connection Name: Excel_GPR
- Connection Type: Odbc
- ODBC Source: Excel_GPR
- Work Directory: c:\TEMP
- Extract User ID: (empty)
- Password: (empty)
- Administrator User ID: (empty)
- Password: (empty)
- Default Load Type: Native ODBC
- SSIS Connection String: Externally loaded, Integration Services load, Native ODBC (highlighted with a red circle), ODBC load, (Default)
- Data Type Mapping Set: (empty)
- Reset button

The load table will look somewhat like this, where an individual Sheet1\$ is used as a source. The white spaces in columns name have to be resolved by using square brackets around the names.

Column Name	Display Name	Data Type	Source Table	Source Column
date	date	varchar(255)	Sheet1\$	Date
start_date	start date	varchar(255)	Sheet1\$	Start Date
office	office	varchar(255)	Sheet1\$	Office
cons#_file_#	cons# file #	numeric(15)	Sheet1\$	Cons# File #
consultant	consultant	varchar(255)	Sheet1\$	Consultant
client_order_#	client order #	numeric(15)	Sheet1\$	Client Order #
co#	co#	varchar(255)	Sheet1\$	[CO#]
rec_id#	rec id#	varchar(255)	Sheet1\$	REC ID#
rec	rec	varchar(255)	Sheet1\$	REC
[1099_com#]	1099 com#	varchar(255)	Sheet1\$	1099 Com#

When Integration Services Load option is chosen then the SSIS Connection String will be filled in once Reset is pressed. SSIS Connect String is a valid SSIS entry that can be used to connect to the data source or destination. The Reset button will attempt to construct a valid connection string from the connection information supplied in the Connection's details. This string consists of the Database ID, Database Link ID (Instance name), Provider Name, Extract User details and more. The string's parameters can be customized. WhereScope will generate a SSIS package with all the functionality provided by Microsoft software stock.

Example of a SSIS Connect String:

```
Provider=OraOLEDB.Oracle.1;Data Source=SOURCE;
Persist Security Info=True;Password=PASSWD;User ID=USER;Auto Translate=False;
```

The screenshot shows the 'Excel_GPR Definition' window with the 'Connection Properties' tab selected. The 'Data Warehouse' checkbox is unchecked. The 'Connection Name' is 'Excel_GPR', 'Connection Type' is 'Odbc', and 'ODBC Source' is 'Excel_GPR'. The 'Work Directory' is 'c:\TEMP'. The 'Extract User ID' and 'Administrator User ID' fields are empty, with corresponding password fields. The 'Default Load Type' is set to 'Integration Services load', which is circled in red. The 'SSIS Connection String' is 'DSN=Excel_GPR;Auto Translate=False;', and the 'Data Type Mapping Set' is '(Default)'. A 'Reset' button is visible on the right.

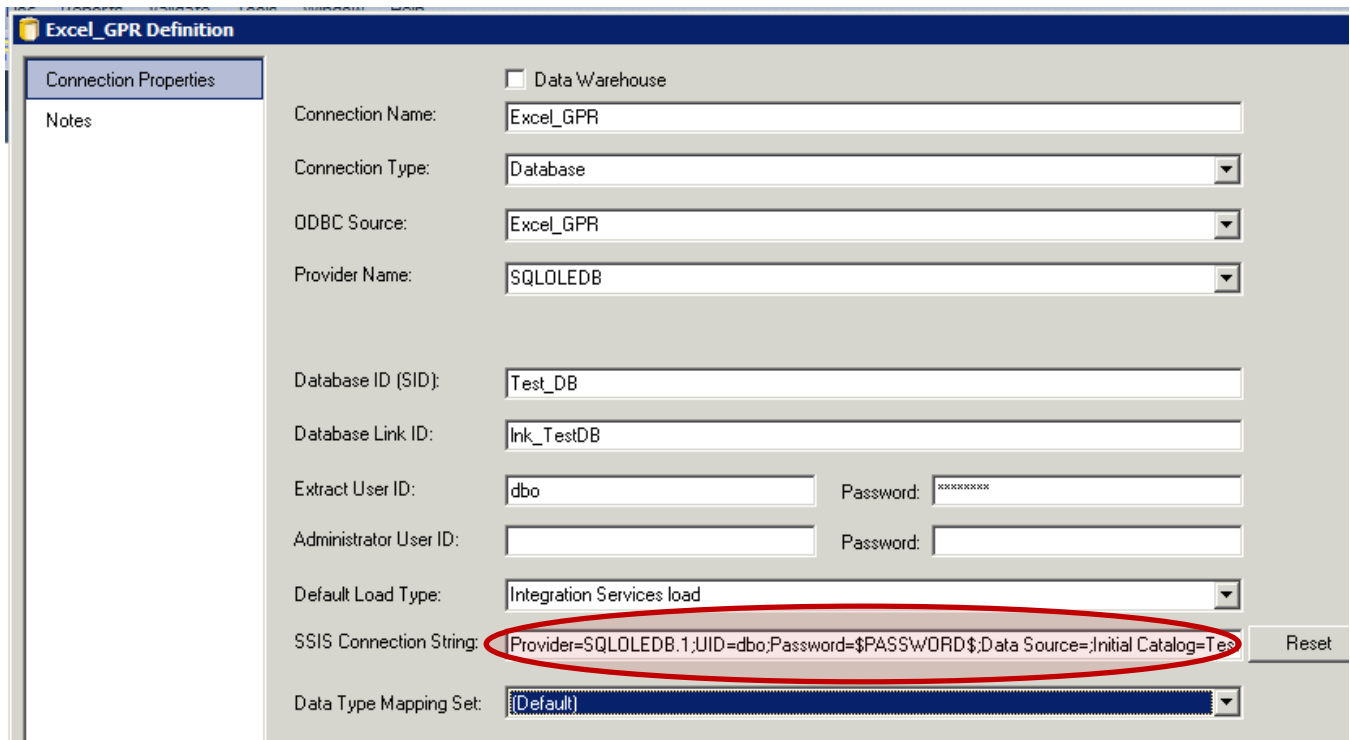
If “ODBC load” is used, then load will be done on row-by-row basis (ODBC Load - Single Row Inserts). This is the most “forgiving”, yet the slowest method, which allows to partial or sample loading

2. RDBMS-based Load

This is very common to use DB-based loading method with Excel loading because it provides connectivity via database links, a very effective and fast technique. The following screen illustrates the Database connection; Excel file can be loaded using a database link loading through a linked server with use of Microsoft.Jet.OLEDB.4.0 provider.

The screenshot shows the 'Excel_GPR Definition' window with the 'Connection Properties' tab selected. The 'Data Warehouse' checkbox is unchecked. The 'Connection Name' is 'Excel_GPR', 'Connection Type' is 'Database', and 'ODBC Source' is 'Excel_GPR'. The 'Provider Name' is 'Microsoft.Jet.OLEDB.4.0', which is circled in red. The 'Full Database Path Name' is 'c:\TEMP', with a 'Browse' button next to it. The 'Database ID (SID)' is 'Test_DB' and the 'Database Link ID' is 'lnk_Test_DB'. The 'Extract User ID' is 'dbo' and the 'Administrator User ID' is empty, with corresponding password fields. The 'Default Load Type' is set to 'Database link load', which is circled in red. The 'SSIS Connection String' is empty, and the 'Data Type Mapping Set' is '(Default)'. A 'Reset' button is visible on the right.

Similarly to the the ODBC-based load, the Database connection type can too have an option of “Integration Services Load”. Note the use of parameters in the SSIS Connection String (more on parameterized load further).

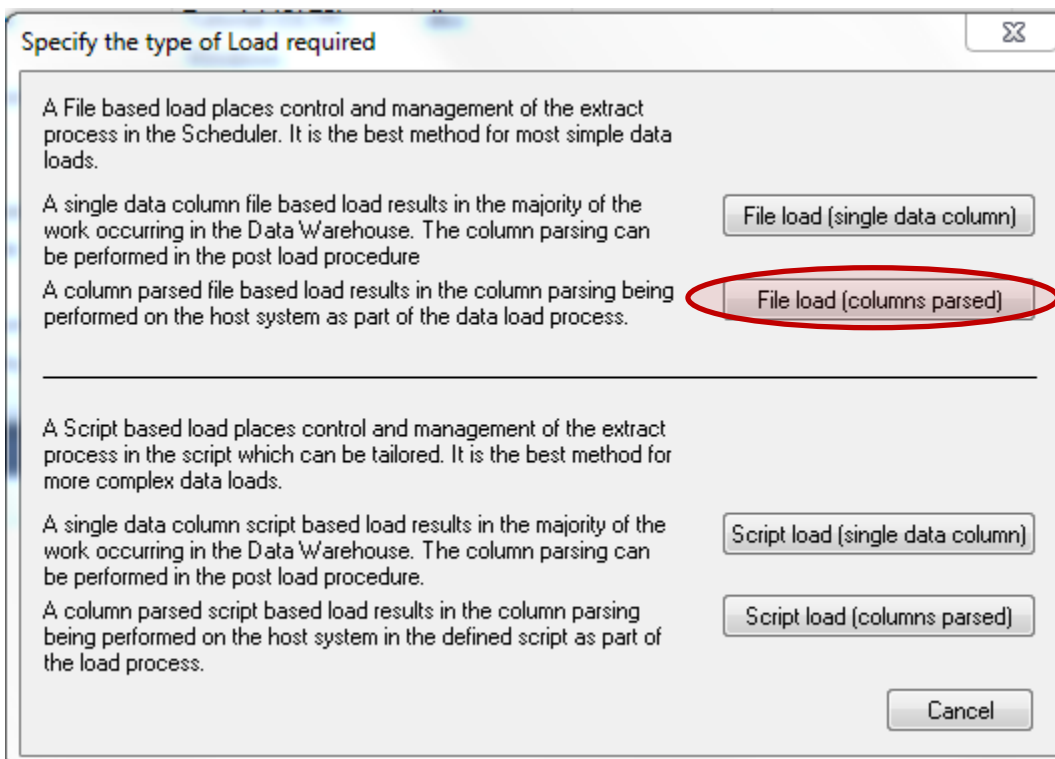


The image shows the 'Excel_GPR Definition' dialog box. It has a left sidebar with 'Connection Properties' and 'Notes'. The main area contains various fields for configuring a database connection. A red circle highlights the 'SSIS Connection String' field, which contains the text: 'Provider=SQLOLEDB.1;UID=dbo;Password=\$PASSWORD\$;Data Source=;Initial Catalog=Test'. Other fields include 'Connection Name' (Excel_GPR), 'Connection Type' (Database), 'ODBC Source' (Excel_GPR), 'Provider Name' (SQLOLEDB), 'Database ID (SID)' (Test_DB), 'Database Link ID' (lnk_TestDB), 'Extract User ID' (dbo), 'Administrator User ID' (empty), 'Default Load Type' (Integration Services load), and 'Data Type Mapping Set' ((Default)). A 'Reset' button is located at the bottom right.

Field	Value
Connection Name	Excel_GPR
Connection Type	Database
ODBC Source	Excel_GPR
Provider Name	SQLOLEDB
Database ID (SID)	Test_DB
Database Link ID	lnk_TestDB
Extract User ID	dbo
Administrator User ID	
Default Load Type	Integration Services load
SSIS Connection String	Provider=SQLOLEDB.1;UID=dbo;Password=\$PASSWORD\$;Data Source=;Initial Catalog=Test
Data Type Mapping Set	((Default))

3. File-based Load

To load data from CSV file using this technology, we will use a “Windows” connection that is provided by Wherescape out-of-the-box. When the option “File Load (columns parsed)” is chosen, Wherescape will guide you through the wizard of creating columns, providing names and data types as well as option for data conversion.



The image shows the 'Specify the type of Load required' dialog box. It contains three sections of text describing different load types: File based load, Script based load, and a section for File load (columns parsed) which is circled in red. The 'File load (columns parsed)' button is also circled in red. Other buttons include 'File load (single data column)', 'Script load (single data column)', 'Script load (columns parsed)', and 'Cancel'.

Specify the type of Load required

A File based load places control and management of the extract process in the Scheduler. It is the best method for most simple data loads.

A single data column file based load results in the majority of the work occurring in the Data Warehouse. The column parsing can be performed in the post load procedure

A column parsed file based load results in the column parsing being performed on the host system as part of the data load process.

File load (columns parsed)

A Script based load places control and management of the extract process in the script which can be tailored. It is the best method for more complex data loads.

A single data column script based load results in the majority of the work occurring in the Data Warehouse. The column parsing can be performed in the post load procedure.

A column parsed script based load results in the column parsing being performed on the host system in the defined script as part of the load process.

Script load (columns parsed)

Cancel

Data load Wizard - Column Definition

Column Data:	File
employee_num	employee_num,cost_center,iscontractor
1230	1230,998,No
1231	1231,997,Yes
1232	1232,997,No
1233	1233,998,No
1234	1234,998,No
1235	1235,998,No
1236	1236,996,Yes
1237	1237,996,Yes
1238	1238,996,Yes
1239	1239,999,No

Display decimal character values

Column Name:

Business Display Name:

Data Type: ☒ Nulls

Conversion:

The Script based load provides endless possibilities to apply various business rules and programming logic while loading Excel file into load table. Those business logics may include removing/reformatting embedded text or images from a spreadsheet; looping through multiple sheets or multiple workbooks while loading data; using wild cards and many more...

4. Parameterized Approach to an Excel Loading

Parameters are a means of passing information between two or more procedures and/or between the WhereScape RED environment and procedures. They can be easily edited within the WhereScape RED framework. Parameters are widely used for loading in general and Excel -- in particular.

Very frequently the parameters are used in incremental loading, an example shown

Where and Group By Clauses:
Parameters can be replaced inline by entering them in the following construct \$Pparameter_name\$. (i.e. leading \$P and trailing \$)

```
WHERE source_tbl.curent_dt
BETWEEN CONVERT(datetime, '$PLast_run_dt$')
AND CONVERT(datetime, '$PCurrent_dt$')
OR '$PFull_load_flag$' = 'Y'
```

Parameters:

Current_Dt = Timestamp of current business date, updates programmatically

Last_Run_Dt = Timestamp of last run date, updates programmatically

Full_load_flag = Indicates Full or incremental load

Another use for parameters in loading process:

load_gpr Table Definition

Properties

File Attributes

Storage

File Path:

File Name:

File Column Delimiter: CHAR(nn) inserts an ascii character (e.g. CHAR(9) = tab)

Parameter: FEED_DIRECTORY=Feed directory that varies depending on environment

Below is a sample code that illustrates WhereScape's ability to utilize parameters to dynamically load multiple Excel workbooks with multiple tabs.

```
BEGIN
-- Select from reference table that holds info about the Excel files designated for loading

SELECT @v_count=count(*)
FROM XLS_PROC_REF WHERE ISNULL(xl_status,'I')='A'

IF @v_count=0

BEGIN
    SET @v_msgtext='There are no excel files to load.'
END
ELSE

BEGIN
    WHILE @v_count <> 0

    BEGIN

        -- Set the lock if wish. Execute wherescape callable procedure that allow to write a parameter

EXEC dbo.WsParameterWrite 'processlockflag','Y','Set to Y if a load process is running.'

        -- Get the first workbook to process
        -- Some parameters used in the code below:
            -- XL_FILE=Excel File (workbook) name
            -- XL_DIR=Directory where Excel file located
            -- XL_SHEET=Excel Tab (worksheet) name, e.g. ['1101$']
            -- XL_USER=Username to be used for extraction Excel Files
            -- XL_PWORD=Password to be used for extraction Excel Files

SELECT top 1 @v_xl_dir=xl_dir,
@v_xl_file=REPLACE(xl_dir COLLATE Latin1_General_BIN,
                    '\\fccokfil001\fccdata\accounting\Common\ACCTG\','Y:\ACCTG\'),
@v_xl_sheet=xl_sheet, @v_xl_user=xl_user, @v_xl_pword=xl_pword, @v_fc_sub=fc_sub
FROM XLS_PROC_REF WHERE ISNULL(xl_status,'I')='A'

        -- Build the sql statement and execute it, using OPENDATASOURCE method
        -- This sql statement varies depending on XL_FILE type, i.e. Sub
        -- LOGIC as follows:

IF @v_fc_sub = 'SUB4'

BEGIN
SET @v_sql='INSERT INTO load_COLL_ACCT SELECT '
SET @v_sql+= @v_xl_dir+', '+@v_xl_file+', '+@v_xl_sheet+', '+@v_xl_user+', '+@v_xl_pword
SET @v_sql+= ',F1,F2,F3,F4,F5,F6,F7,F8,F9,F10,'+CONVERT(varchar,@v_dss_update_time, 9)
SET @v_sql+= ' FROM OPENDATASOURCE(''Microsoft.ACE.OLEDB.12.0'',
''Data Source=@v_xl_dir +''\'+ @v_xl_file
SET @v_sql+= char(59)+'user id=@v_xl_user'
SET @v_sql+= char(59)+'password=@v_xl_pword'
SET @v_sql+= char(59)+'Extended Properties=Excel 12.0'' )'
SET @v_sql+= '... ['''+@v_xl_sheet+'$'' ]'
END

ELSE IF @v_fc_sub = 'SUB5'
BEGIN
```

```

SET @v_sql='INSERT INTO load_COLL_ACCT SELECT '
SET @v_sql+= @v_xl_dir+', '+@v_xl_file+', '+@v_xl_sheet+', '+@v_xl_user+', '+@v_xl_pword
SET @v_sql+= ',F1,F2,F3,F4,F5,F6, '+CONVERT(varchar, @v_dss_update_time, 9)
SET @v_sql+= ' FROM OPENDATASOURCE(''Microsoft.ACE.OLEDB.12.0'',
        ''Data Source=c:\testing\''+@v_xl_file
SET @v_sql+= char(59)+'user id=@v_xl_user'
SET @v_sql+= char(59)+'password=@v_xl_pword'
SET @v_sql+= char(59)+'Extended Properties=Excel 12.0''')'
SET @v_sql+= '... [\'\''+@v_xl_sheet+'$\'\'']'
END

exec sp_executesql @v_sql

UPDATE XLS_PROC_REF
SET xl_status='I'
WHERE xl_dir=@v_xl_dir AND xl_file=@v_xl_file AND xl_sheet=@v_xl_sheet
        AND xl_status='A' AND fc_sub=@v_fc_sub

SELECT @v_count=count(*)
FROM XLS_PROC_REF WHERE ISNULL(xl_status,'I')='A'
END
END

```