**Supplier Conversion Using Interface Approach**

To Create Supplier using Interface approach, first we need supplier data (flat files) and staging tables and one package with validations and Control files to Insert data into the staging tables.

**Example for Creating Supplier using Interface Approach.**

In this Example I’m creating Supplier, Supplier Site and Contact for Supplier.

First, we need flat files for creating Supplier.







After creating flat file or receiving the flat files from the client, we need to prepare Control files for each Falt file.

And the file extension must be ( **.ctl** ).

**The structure of the control file is**

➔ **Options** is used to precedes the load data

**Options Parameters are**

**SKIP** = n – Number of logical records to skip (Default 0)

**LOAD** = n — Number of logical records to load (Default all)

**ERRORS** = n — Number of errors to allow (Default 50)

**ROWS** = n — Number of rows in conventional path bind array or between direct path data saves (Default: Conventional Path 64, Direct path all)

**BINDSIZE** = n – Size of conventional path bind array in bytes (System-dependent default)

**SILENT** = {FEEDBACK | ERRORS | DISCARDS | ALL} — Suppress messages during run (header, feedback, errors, discards, partitions, all)

**DIRECT** = {TRUE | FALSE} –Use direct path (Default FALSE)

**PARALLEL** = {TRUE | FALSE} — Perform parallel load (Default FALSE)

➔**LOADDATA** statement is required at the beginning of the control file.

➔**INFILE** is used to hold the data file in the control file

After INFILE select How to insert data into the table

After INFILE we can use these following loading types

**INSERT** — If the table you are loading is empty, INSERT can be used.

**APPEND** — If data already exists in the table, SQL\*Loader appends the new rows to it. If data doesn’t already exist, the new rows are simply loaded.

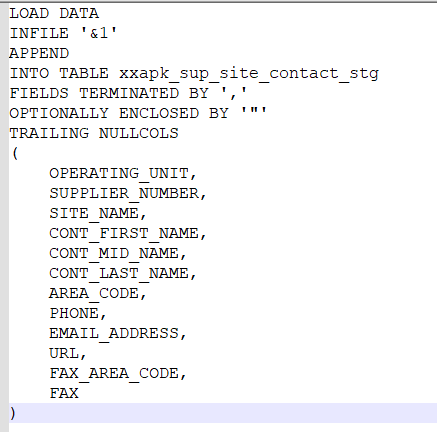
**REPLACE** — All rows in the table are deleted and the new data is loaded

**TRUNCATE** — SQL\*Loader uses the SQL TRUNCATE command.

➔**INTO TABLE** It is required to mention to which table we want to insert data.

➔**FIELDS TERMINATED** It specify how the data fields are terminated in the data file(delimiter) After this mention your table column names

**Ex: -**



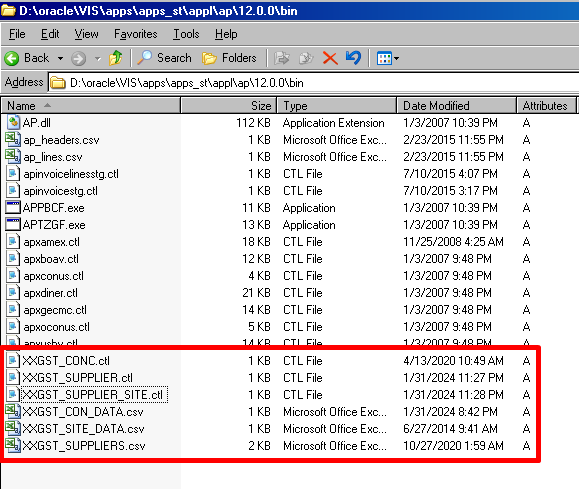


Above are the control files according to the flat files.

After creating control files, we need to place all these files (**Control files and Flat files**) in one location to run the concurrent programs.

So, here I’m placing the file in **AP top (**Account Payables**).**

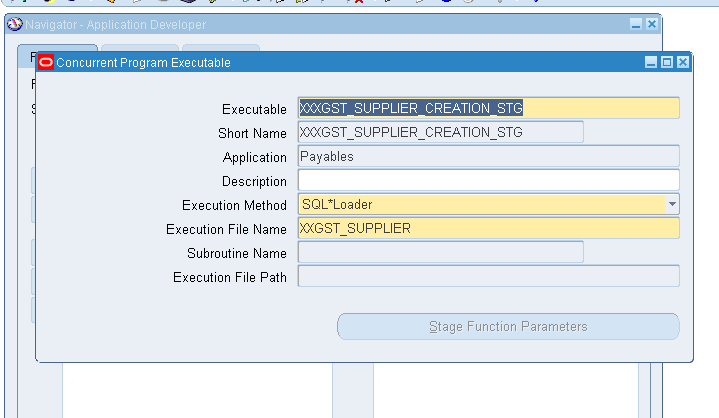
Path: - **D:\oracle\VIS\apps\apps\_st\appl\ap\12.0.0\bin**



After placing the file in one location, we need to create Concurrent programs to run the control files

With Execution method as **SQL\*Loader.**

* **Executable for insert Supplier into Staging table.**

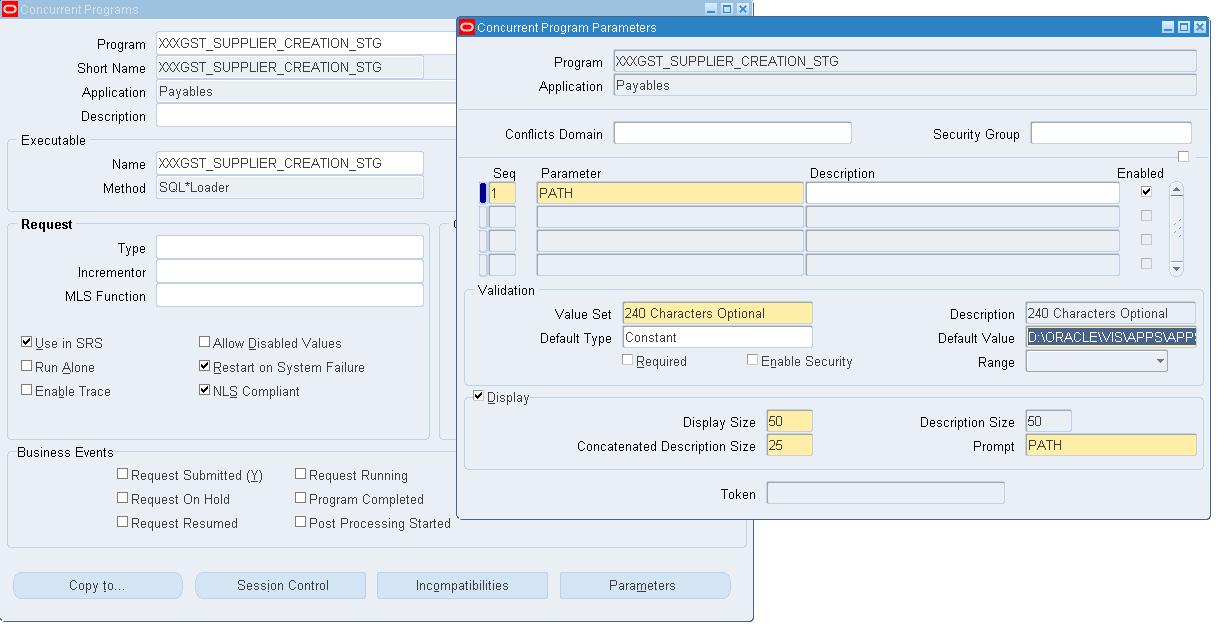


Here, Execution file name is **Control file name** without extension**.**

Make sure the Application type is same as where you placed control files and flat files.

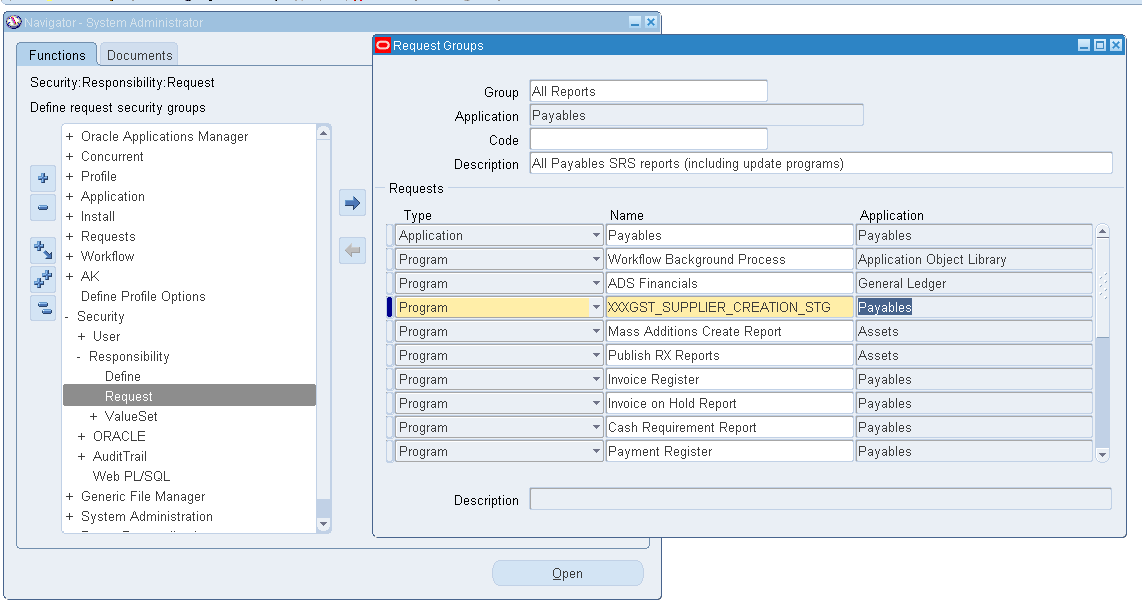
In this case I placed the files in AP top, So I selected Application as a “**Payables”**.

* **Concurrent Program for above executable.**



Here, we need to define one parameter to give the flat file path to run the SQL\*Loader program.

Example for path: **D:\oracle\VIS\apps\apps\_st\appl\ap\12.0.0\bin\** **XXGST\_SUPPLIERS.csv**

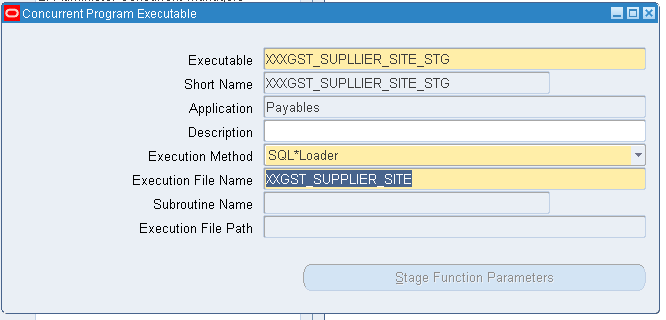


* After creating the program, we need to assign this program to a request group.

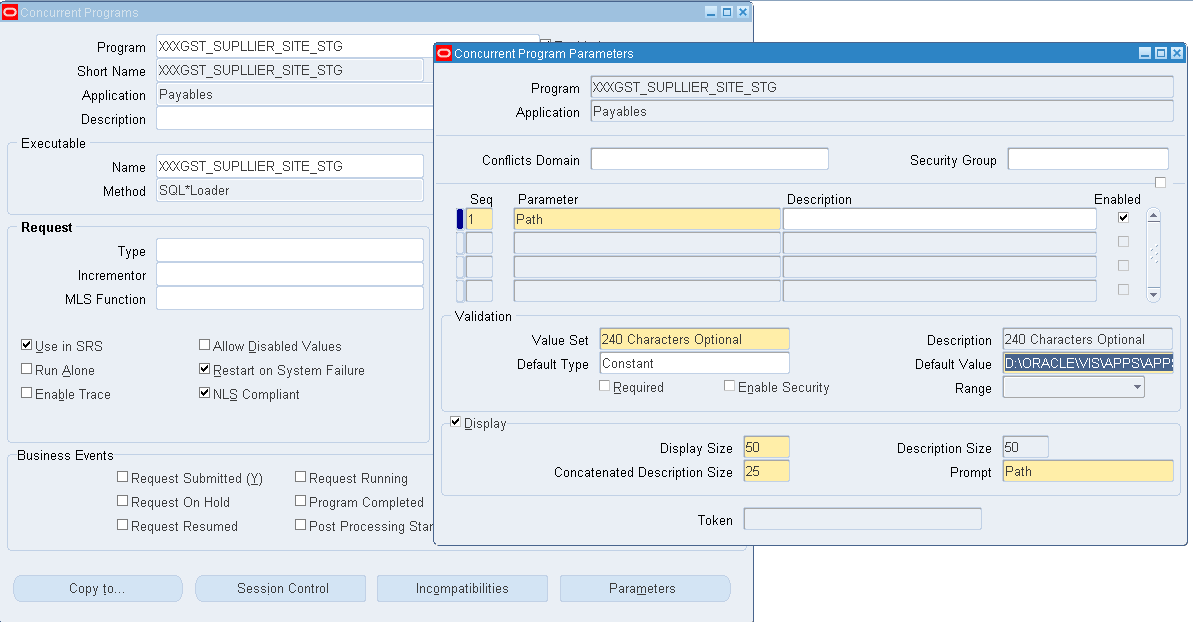
Concurrent Program successfully created for **Supplier.**

Same process for **Supplier site** and **Contact.**

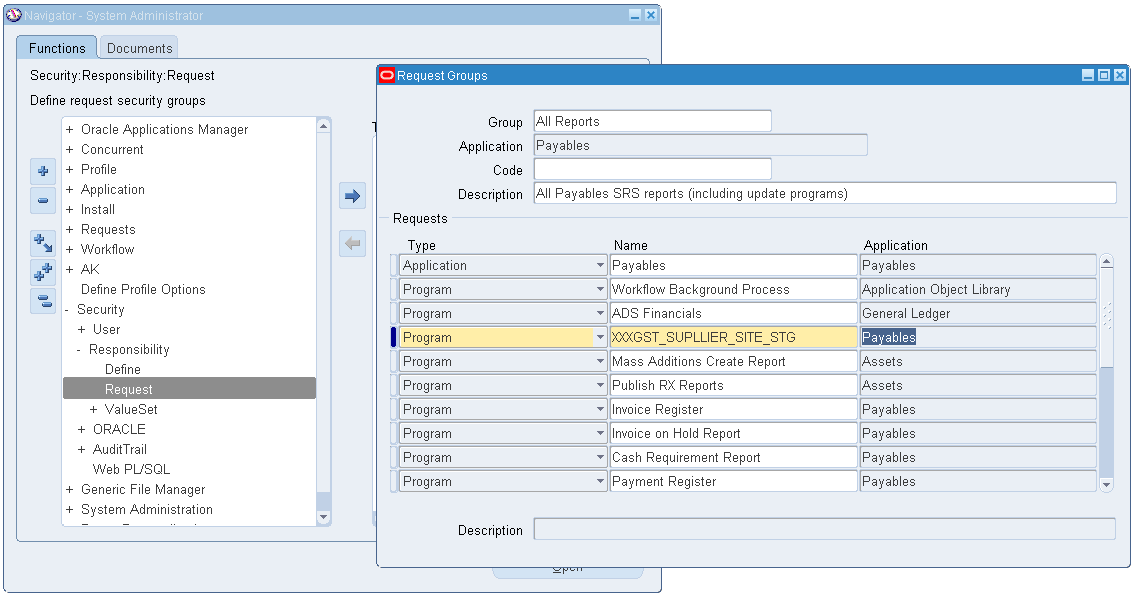
* **Executable for Supplier Site.**



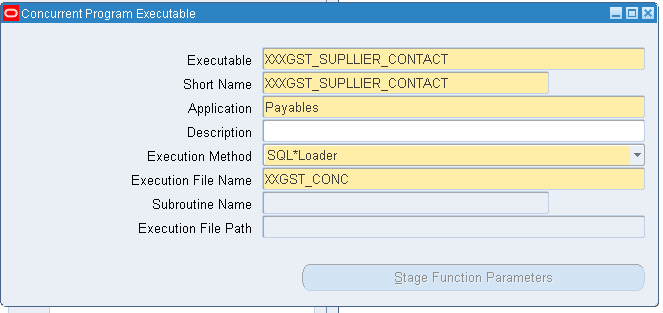
* **Concurrent Program for Supplier Site**



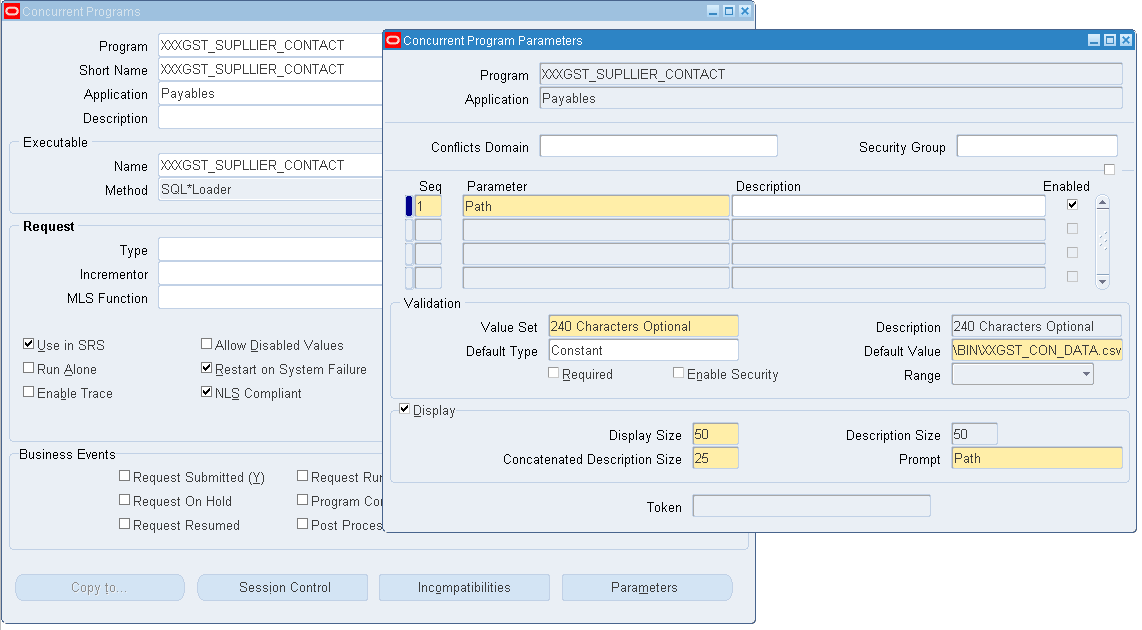
* **Adding Concurrent program to a request group.**



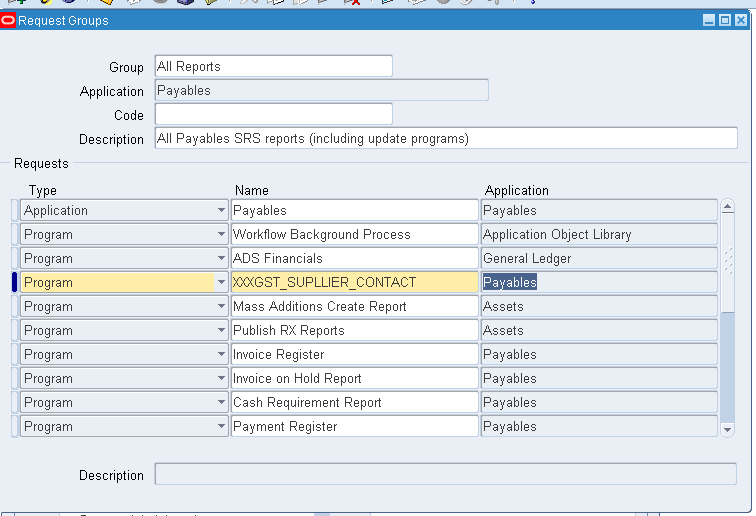
* **Executable for Supplier Contact.**



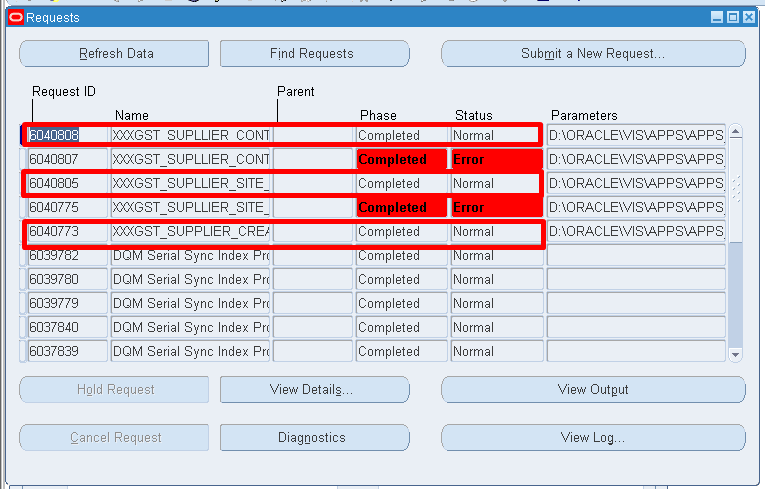
* **Concurrent program for Supplier Contact.**



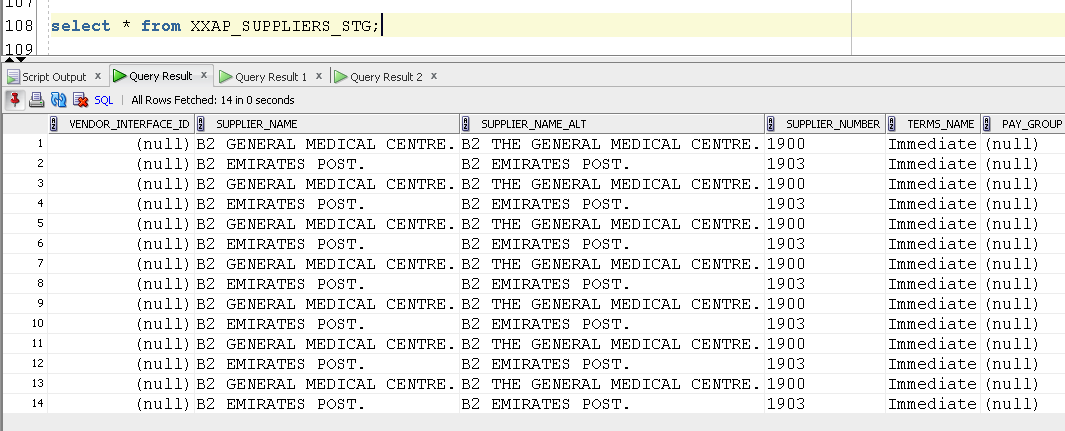
* **Adding Concurrent program to a request group.**

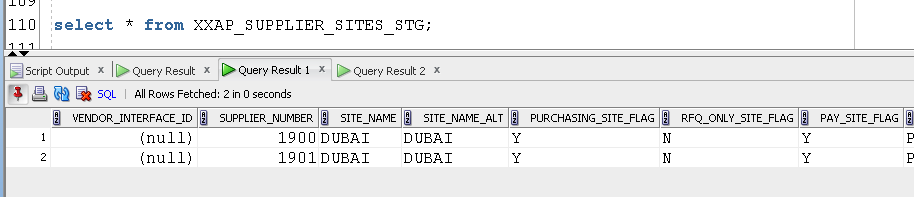


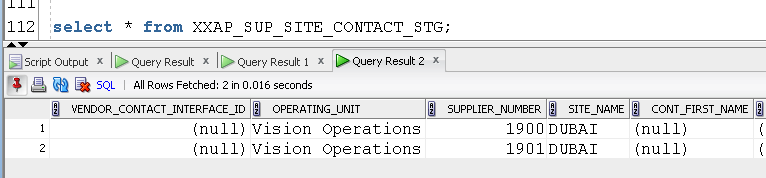
* **Submitting the Concurrent Programs**



**Check data is successfully inserted into staging table or not.**







* **After data inserted into staging table, we need to create one package for validations and**

**Transfer data from staging table to interface table.**

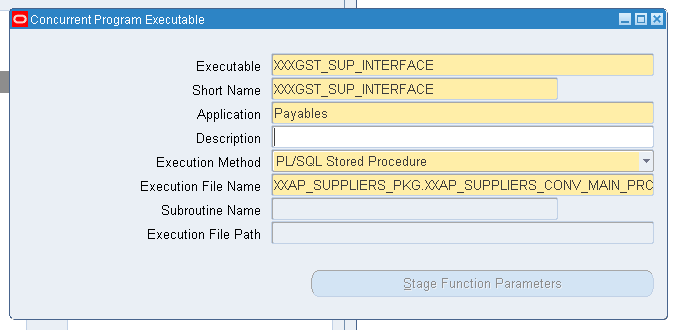
**Package: -**



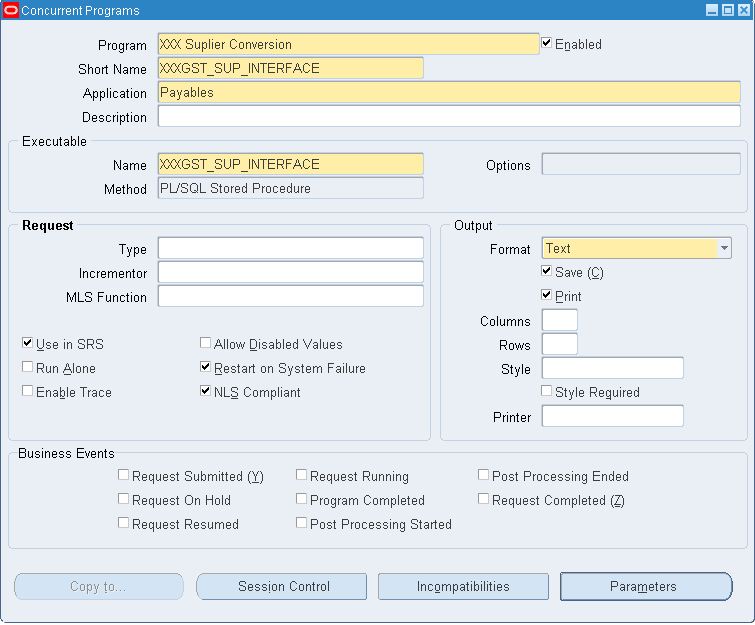
* **After Compiling the package, we need to create Concurrent Program for that package.**
* **Executable for the package**

Execution Method as **PL/SQL Stored Procedure.**

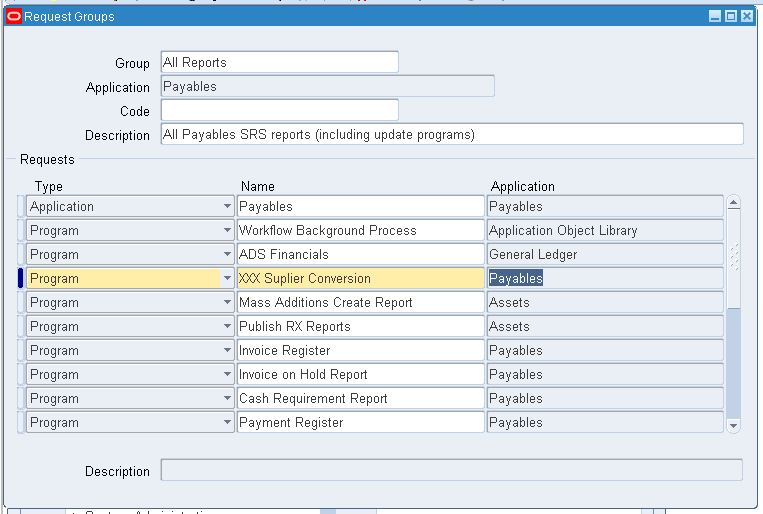
Here Execution file name is **Package.Procedure**.



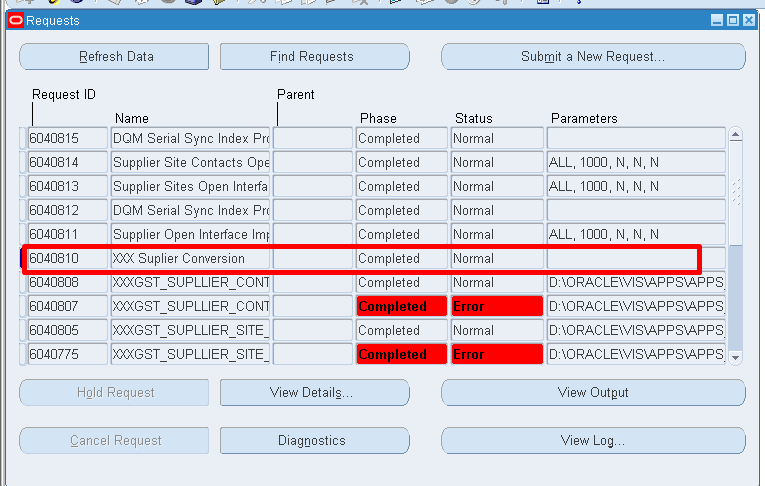
* **Concurrent Program for above executable.**



* **Adding Concurrent Program to a Request Group.**



* **Submit the Concurrent Program, the data will transfer from staging table to Interface tables.**

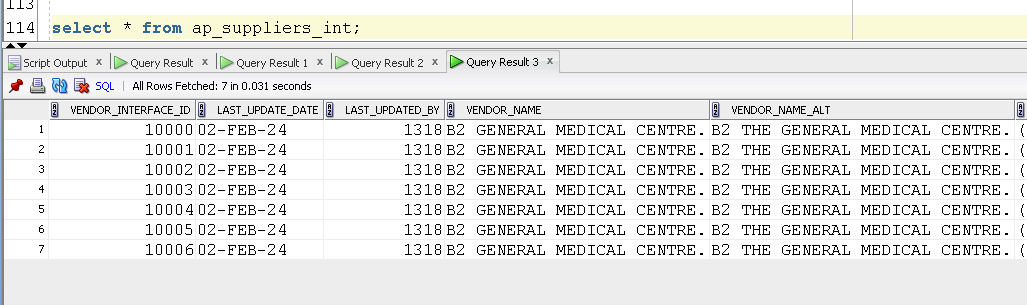


Now, we need to check the interface tables.

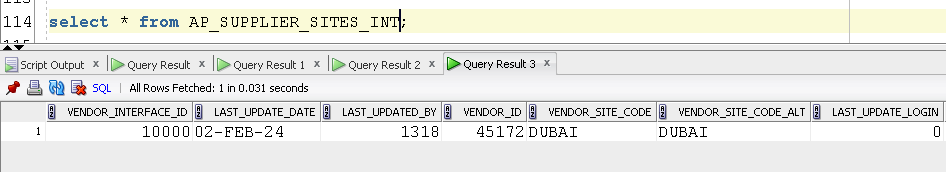
The Interface tables are

**AP\_SUPPLIERS\_INTAP\_SUPPLIER\_SITES\_INTAP\_SUP\_SITE\_CONTACT\_INT**

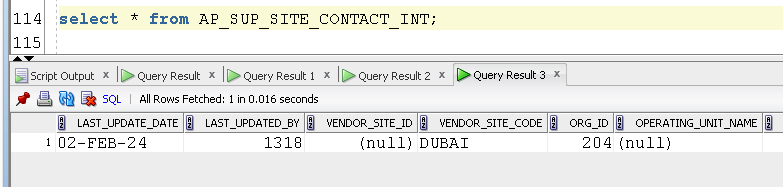
**AP\_SUPPLIERS\_INT**

****

**AP\_SUPPLIER\_SITES\_INT**

****

**AP\_SUP\_SITE\_CONTACT\_INT**

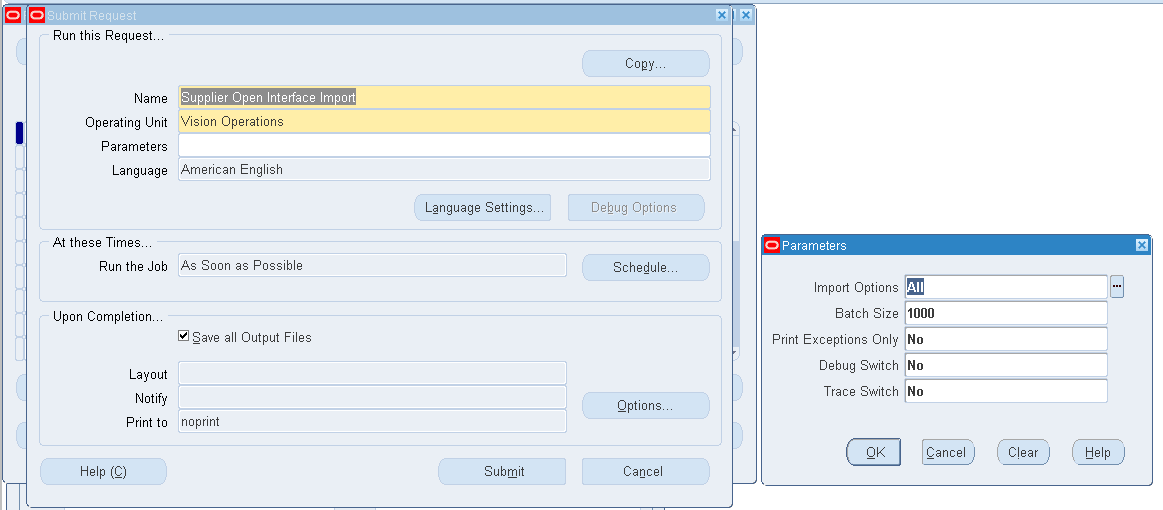
****

* **Now we need to run the seeded Concurrent programs to transfer data from Interface table to Base table.**

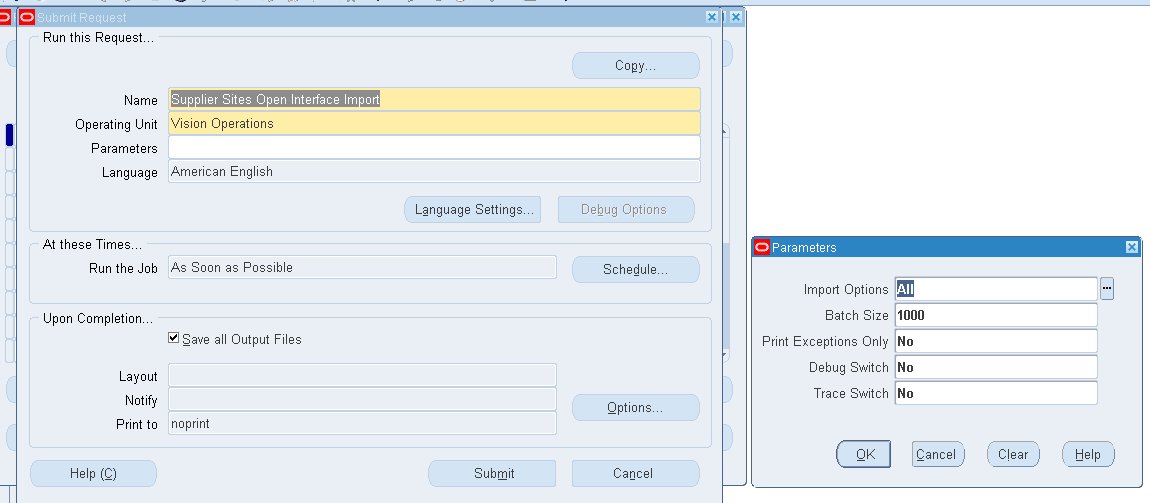
**The Standard Programs are**

1. **Supplier Open Interface Import**
2. **Supplier Sites Open Interface Import**
3. **Supplier Site Contacts Open Interface Import**

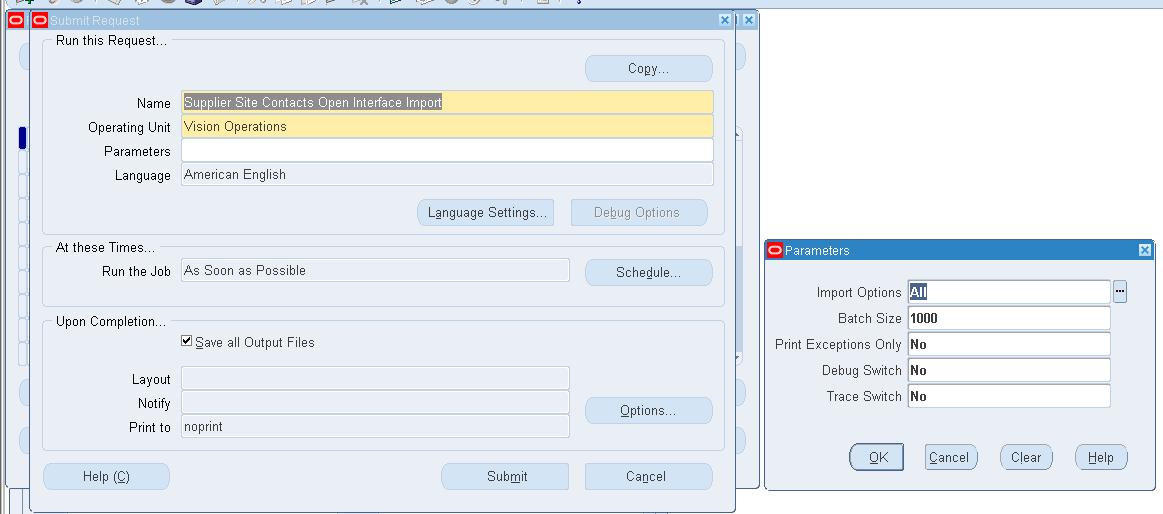
**Supplier Open Interface Import**

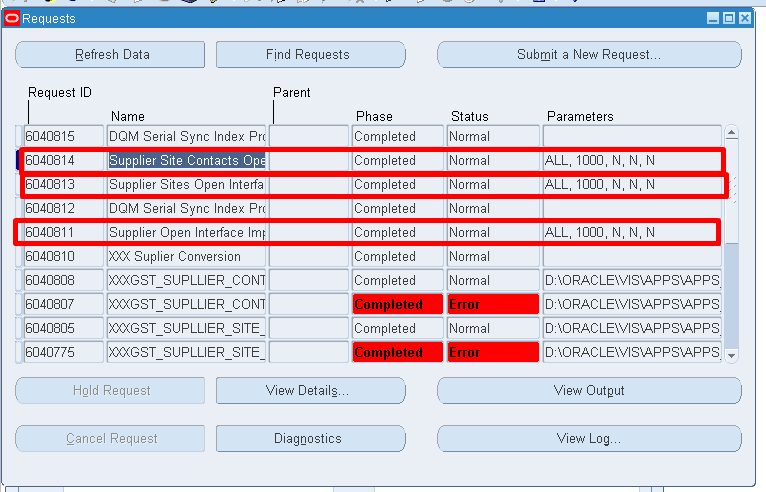


**Supplier Sites Open Interface Import**

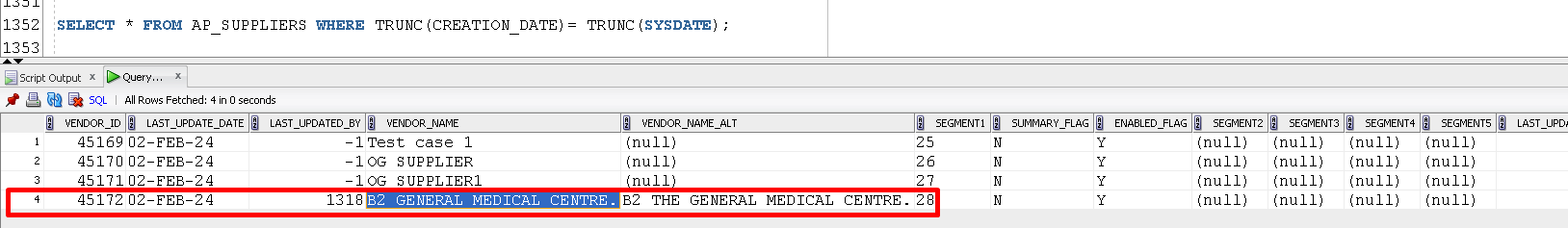


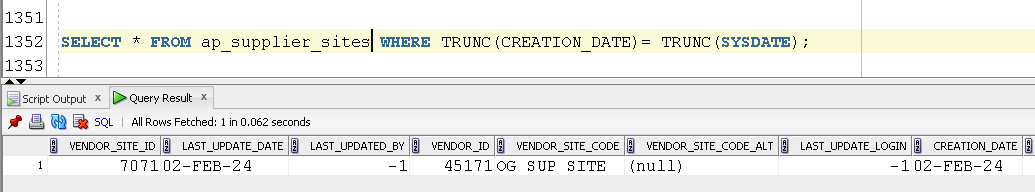
**Supplier Site Contacts Open Interface Import**





* **Now we need to check the base table.**





* There are some setup issues in VM Ware. So the Contact is not created.

