

AR Invoice - Interface

- ▶ Interface Type : Public API
- ▶ Interface Table : Not Available
- ▶ Public API : AR_INVOICE_API_PUB.create_single_invoice
- ▶ Base Table : ra_customer_trx_all, ra_customer_trx_lines_all
- ▶ Error Table : ar_trx_errors_gt
- ▶ Import Job : Not Available

Implementation Process

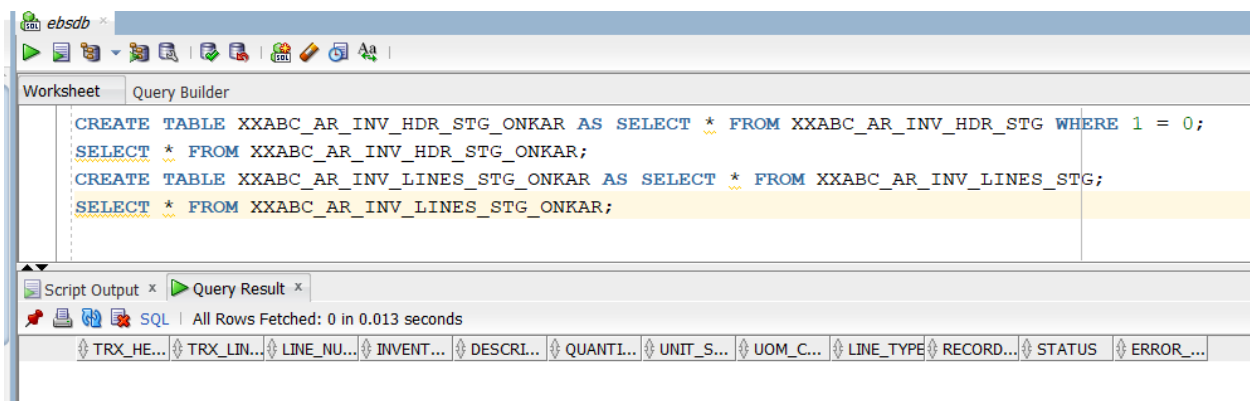
- ▶ Designed an Unix Shell Script to invoke the below SQLLoader Programs
 - ▶ Designed a SQL Loader Programs (Header, Lines) to read data files and upload to staging tables (XXABC_AR_INV_HDR_STG, XXABC_AR_INV_LINES_STG)
 - ▶ *Header Data File (XXABC_INV_HDR_DATA.csv)*
 - ▶ *Line Data File (XXABC_INV_LINES_DATA.csv)*
- ▶ Designed a PLSQL API (XXABC_AR_INV_IMP_UTILS) to perform the below functionality
 - ▶ Validate the data in staging table
 - ▶ Invoke the Public API (AR_INVOICE_API_PUB.create_single_invoice)

List of Components

Component Type	Name	Purpose
Table	XXABC_AR_INV_HDR_STG XXABC_AR_INV_LINES_STG	Staging Table
Data File	XXABC_INV_HDR_DATA.csv XXABC_INV_LINES_DATA.csv	AR Invoice Header and Lines
Control File	XXABCARINVDATALOAD.ctf XXABCARINVLINESDATALOAD.ctf	SQLLoader File to load data from Flat file to staging table
PLSQL	XXABC_AR_INV_IMP_UTILS	PLSQL Package to load data into Interface Table and Call Journal Import
Concurrent Program	XXABC AR Invoice Load Files	For Invoking Control file to load two data files
Concurrent Program	XXABC AR Invoice Import	

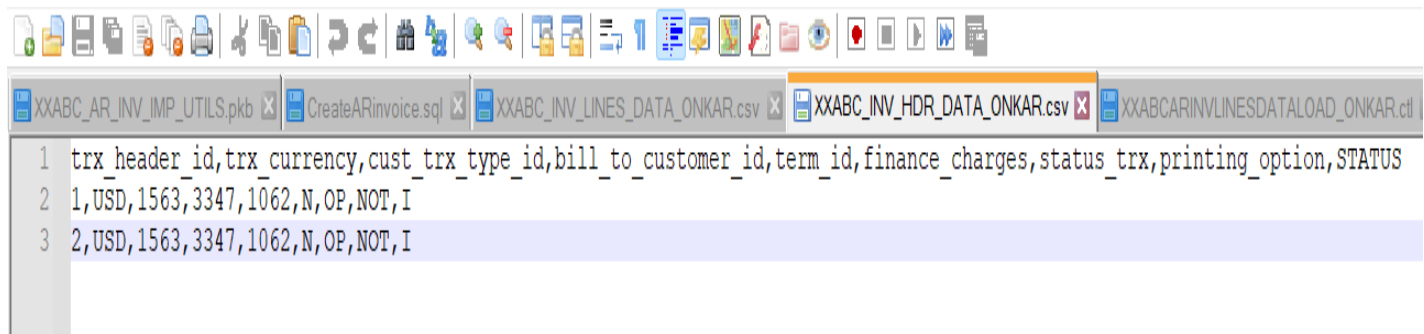
PROCEDURE:

1.CREATE STAGING TABLE:

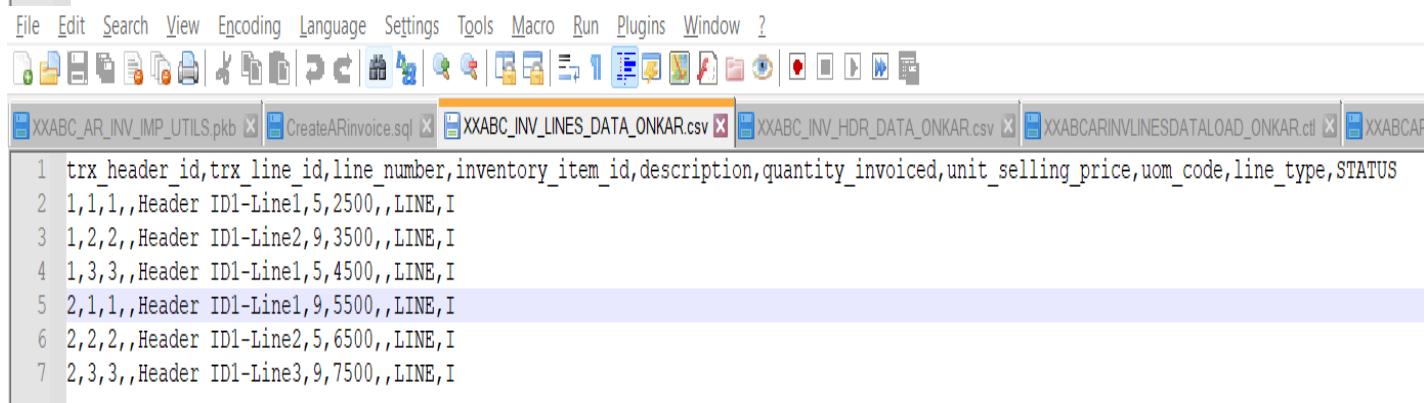


2. SHELL SCRIPT WITH 2 DATAFILE & CONTROL FILES:

DATA IN DATA FILE:

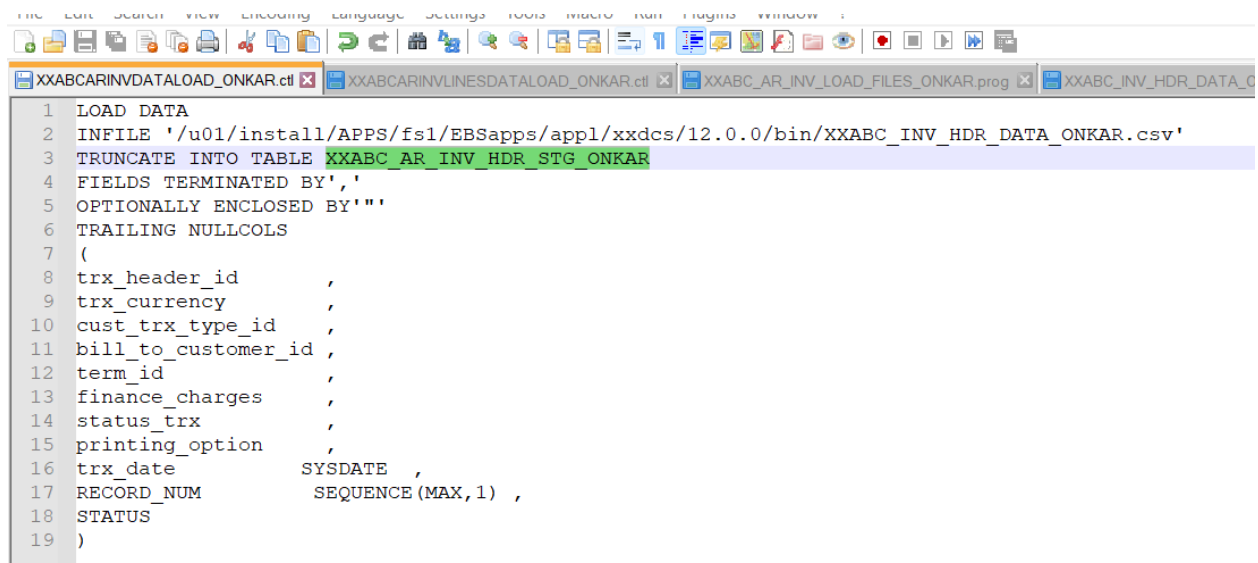


```
1 trx_header_id,trx_currency,cust_trx_type_id,bill_to_customer_id,term_id,finance_charges,status_trx,printing_option,STATUS
2 1,USD,1563,3347,1062,N,OP,NOT,I
3 2,USD,1563,3347,1062,N,OP,NOT,I
```

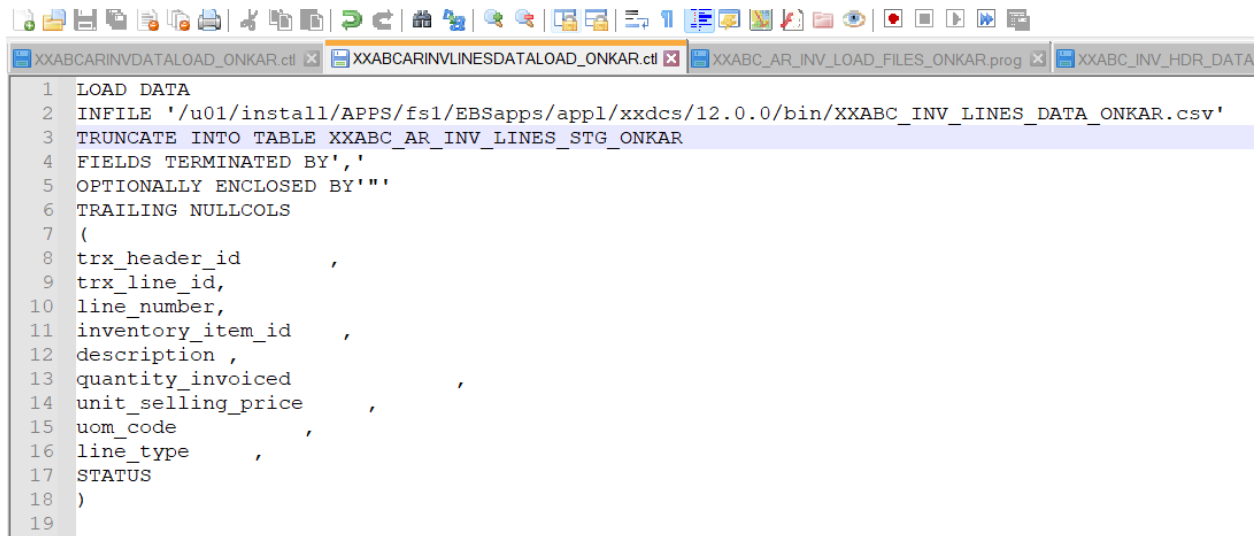


```
1 trx_header_id,trx_line_id,line number,inventory_item_id,description,quantity_invoiced,unit_selling_price,uom_code,line_type,STATUS
2 1,1,1,,Header ID1-Line1,5,2500,,LINE,I
3 1,2,2,,Header ID1-Line2,9,3500,,LINE,I
4 1,3,3,,Header ID1-Line1,5,4500,,LINE,I
5 2,1,1,,Header ID1-Line1,9,5500,,LINE,I
6 2,2,2,,Header ID1-Line2,5,6500,,LINE,I
7 2,3,3,,Header ID1-Line3,9,7500,,LINE,I
```

CTRL FILES:

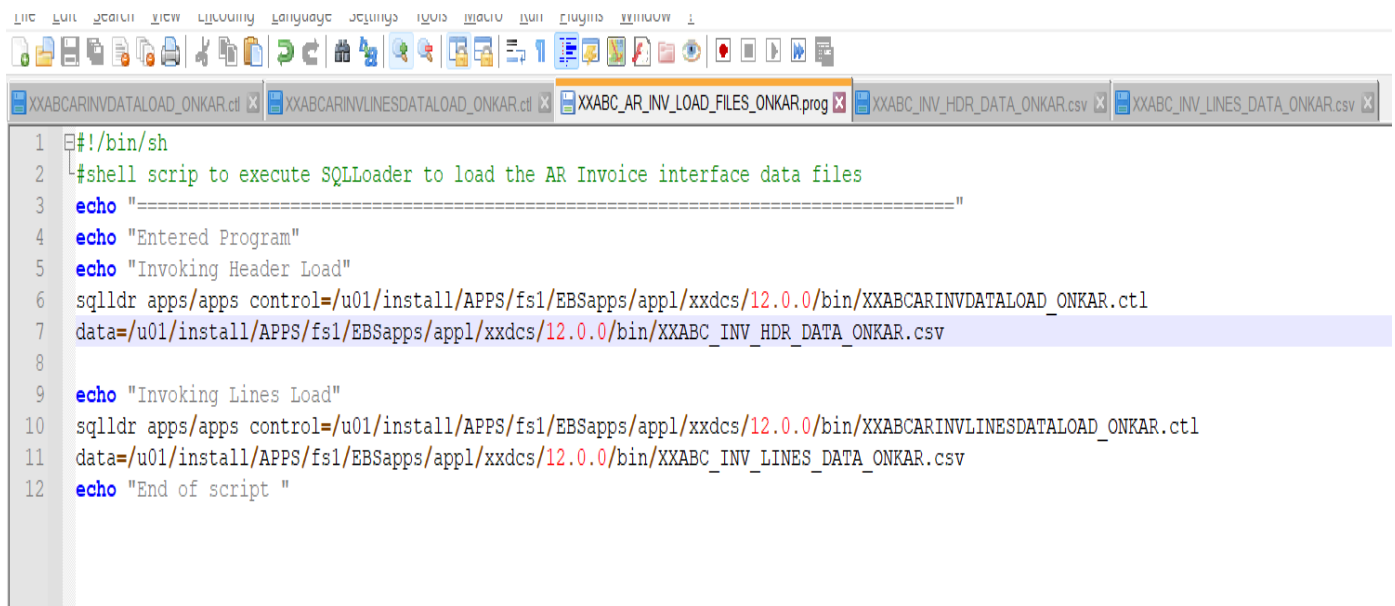


```
1 LOAD DATA
2 INFILE '/u01/install/APPS/fs1/EBSapps/appl/xxdcs/12.0.0/bin/XXABC_INV_HDR_DATA_ONKAR.csv'
3 TRUNCATE INTO TABLE XXABC_AR_INV_HDR_STG_ONKAR
4 FIELDS TERMINATED BY','
5 OPTIONALLY ENCLOSED BY'"'"'
6 TRAILING NULLCOLS
7 (
8   trx_header_id      ,
9   trx_currency       ,
10  cust_trx_type_id   ,
11  bill_to_customer_id ,
12  term_id            ,
13  finance_charges    ,
14  status_trx         ,
15  printing_option     ,
16  trx_date            SYSDATE ,
17  RECORD_NUM          SEQUENCE (MAX,1) ,
18  STATUS
19 )
```



```
1 LOAD DATA
2 INFILE '/u01/install/APPS/fs1/EBSapps/appl/xxdcs/12.0.0/bin/XXABC_INV_LINES_DATA_ONKAR.csv'
3 TRUNCATE INTO TABLE XXABC_AR_INV_LINES_STG_ONKAR
4 FIELDS TERMINATED BY ','
5 OPTIONALLY ENCLOSED BY '"'
6 TRAILING NULLCOLS
7 (
8   trx_header_id      ,
9   trx_line_id,
10  line_number,
11  inventory_item_id  ,
12  description ,
13  quantity_invoiced  ,
14  unit_selling_price ,
15  uom_code           ,
16  line_type          ,
17  STATUS
18 )
19
```

SHELL SCRIPT:



```
1 #!/bin/sh
2 #shell scrip to execute SQLLoader to load the AR Invoice interface data files
3 echo "=====
4 echo "Entered Program"
5 echo "Invoking Header Load"
6 sqlldr apps/apps control=/u01/install/APPS/fs1/EBSapps/appl/xxdcs/12.0.0/bin/XXABCARINVDATLOAD_ONKAR.ct1
7 data=/u01/install/APPS/fs1/EBSapps/appl/xxdcs/12.0.0/bin/XXABC_INV_HDR_DATA_ONKAR.csv
8
9 echo "Invoking Lines Load"
10 sqlldr apps/apps control=/u01/install/APPS/fs1/EBSapps/appl/xxdcs/12.0.0/bin/XXABCARINVLINESDATLOAD_ONKAR.ct1
11 data=/u01/install/APPS/fs1/EBSapps/appl/xxdcs/12.0.0/bin/XXABC_INV_LINES_DATA_ONKAR.csv
12 echo "End of script "
```

3. LOCAL TO SERVER:

Steps to create a symbolic link:

Click on the black icon in the upper left corner and enter the commands(check the spaces in commands)

```
ln -s /u01/install/APPS/fs1/EBSapps/appl/xxdcs/12.0.0/bin/
XXABC_AR_INV_LOAD_FILES_OS.prog
/u01/install/APPS/fs1/EBSapps/appl/xxdcs/12.0.0/bin/
XXABC_AR_INV_LOAD_FILES_OS
```

- This command creates a symbolic (soft) link named **XXDCS_SK1__DATA_LOADER** that points to the file **XXABC_AR_INV_LOAD_FILES_OS.prog**. • A symbolic link allows you to create shortcuts or alternate names for files. You can access **XXABC_AR_INV_LOAD_FILES_OS.prog** by referring to **XXABC_AR_INV_LOAD_FILES_OS**.

```
ls -l /u01/install/APPS/fs1/EBSapps/appl/xxdcs/12.0.0/bin/  
XXABC_AR_INV_LOAD_FILES_OS.prog
```

- This command lists the **XXABC_AR_INV_LOAD_FILES_OS.prog** file in a single column format. • -l specifies single-column output, which is helpful when dealing with multiple items for readability.

```
chmod +x /u01/install/APPS/fs1/EBSapps/appl/xxdcs/12.0.0/bin/  
XXABC_AR_INV_LOAD_FILES_OS.prog
```

- This command changes the file permissions to make **XXABC_AR_INV_LOAD_FILES_OS.prog** executable. • Adding +x gives execute permissions, which is needed if you want to run this file as a script or program.

```
dos2unix /u01/install/APPS/fs1/EBSapps/appl/xxdcs/12.0.0/bin/  
XXABC_AR_INV_LOAD_FILES_OS.prog
```

(command not found error ignore).

- This command converts the file from DOS/Windows format to UNIX format. • DOS/Windows files use CRLF (carriage return + line feed) as a line ending, whereas UNIX uses LF only. dos2unix removes the extra CR to make the file compatible with UNIX systems.

```
sed -i 's/\r$//'  
/u01/install/APPS/fs1/EBSapps/appl/xxdcs/12.0.0/bin/XXABC_AR_INV_LOAD_FILES_  
OS.prog
```

- This command also removes \r (carriage return) characters at the end of each line within the file. • sed is used here to clean up line endings by replacing \r with nothing. This is similar to dos2unix, ensuring the file is in UNIX format.

Name	Size	Type	Changed
..		Parent directory	11/13/2024 10:36:56 AM
XMLOffice		File folder	11/7/2024 3:58:05 PM
sql_plsql_questions_DCS		File folder	11/1/2024 9:41:34 PM
petfam_django_project		File folder	7/4/2024 4:20:25 PM
Oracle_app_tech_consultant_notes_DCS		File folder	11/11/2024 11:52:14 PM
office software		File folder	10/31/2024 12:30:51 PM
XXABCARINVINESDATALOAD_ONKAR.ctl	1 KB	CTL File	11/12/2024 10:52:32 PM
XXABCARINVVDATALOAD_ONKAR.ctl	1 KB	CTL File	11/12/2024 10:51:48 PM
XXABC_INV_LINES_DATA_ONKAR.csv	1 KB	Microsoft Excel Co...	11/12/2024 10:50:09 PM
XXABC_INV_HDR_DATA_ONKAR.csv	1 KB	Microsoft Excel Co...	11/12/2024 10:49:45 PM
XXABC_AR_INV_LOAD_FILES_OS.prog	1 KB	PROG File	11/12/2024 10:58:24 PM
ebs_10_0_10_6.ppk	2 KB	PuTTY Private Key ...	10/17/2024 1:10:17 PM

Name	Size	Changed	Rights	Owner
SK_ItemData		11/7/2024 3:55:33 PM	rwxr-xr-x	oracle
ONKAR_ITEMDATA		11/12/2024 5:06:21 PM	rwxr-xr-x	oracle
AV_ItemData		11/12/2024 12:52:34 PM	rwxr-xr-x	oracle
MI_ItemData		11/11/2024 8:17:31 PM	rwxr-xr-x	oracle
MO_ItemData		11/11/2024 5:04:28 PM	rwxr-xr-x	oracle
MN_ItemData		11/11/2024 5:03:57 PM	rwxr-xr-x	oracle
XXDCS_AR_INV_LOAD_FILES_HH.pkb	7 KB	11/13/2024 11:32:11 AM	rw-r--r--	oracle
XXDCS_AR_INV_LOAD_FILES_HH.prog	1 KB	11/13/2024 11:22:56 AM	rwxr-xr-x	oracle
XXDCS_AR_INV_LOAD_FILES_HH.ctl	1 KB	11/13/2024 11:17:35 AM	rwxr-xr-x	oracle
XXDCS_AR_INV_LINES_DATALOAD_HH.ctl	1 KB	11/13/2024 11:08:16 AM	rw-r--r--	oracle
XXDCS_AR_INV_DATALOAD_HH.ctl	1 KB	11/13/2024 11:07:23 AM	rw-r--r--	oracle
XXDCS_AR_INV_LOAD_FILES_HH.pks	1 KB	11/13/2024 10:57:11 AM	rw-r--r--	oracle
XXDCS_AR_INV_LINES_DATA_HH.csv	1 KB	11/13/2024 10:27:08 AM	rw-r--r--	oracle
XXDCS_INV_HDR_DATA_HH.csv	1 KB	11/13/2024 10:27:08 AM	rw-r--r--	oracle
XXABCARINVINESDATALOAD_ONKAR.ctl	1 KB	11/12/2024 10:52:32 PM	rw-r--r--	oracle
XXABCARINVVDATALOAD_ONKAR.ctl	1 KB	11/12/2024 10:51:48 PM	rw-r--r--	oracle
XXDCS_AR_INV_LOAD_FILES_SUF.prog	1 KB	11/12/2024 5:21:53 PM	rwxr-xr-x	oracle
XXDCS_AR_INV_LOAD_FILES_SUF.ctl	1 KB	11/12/2024 5:19:37 PM	rwxr-xr-x	oracle

Move .csv files in text format.

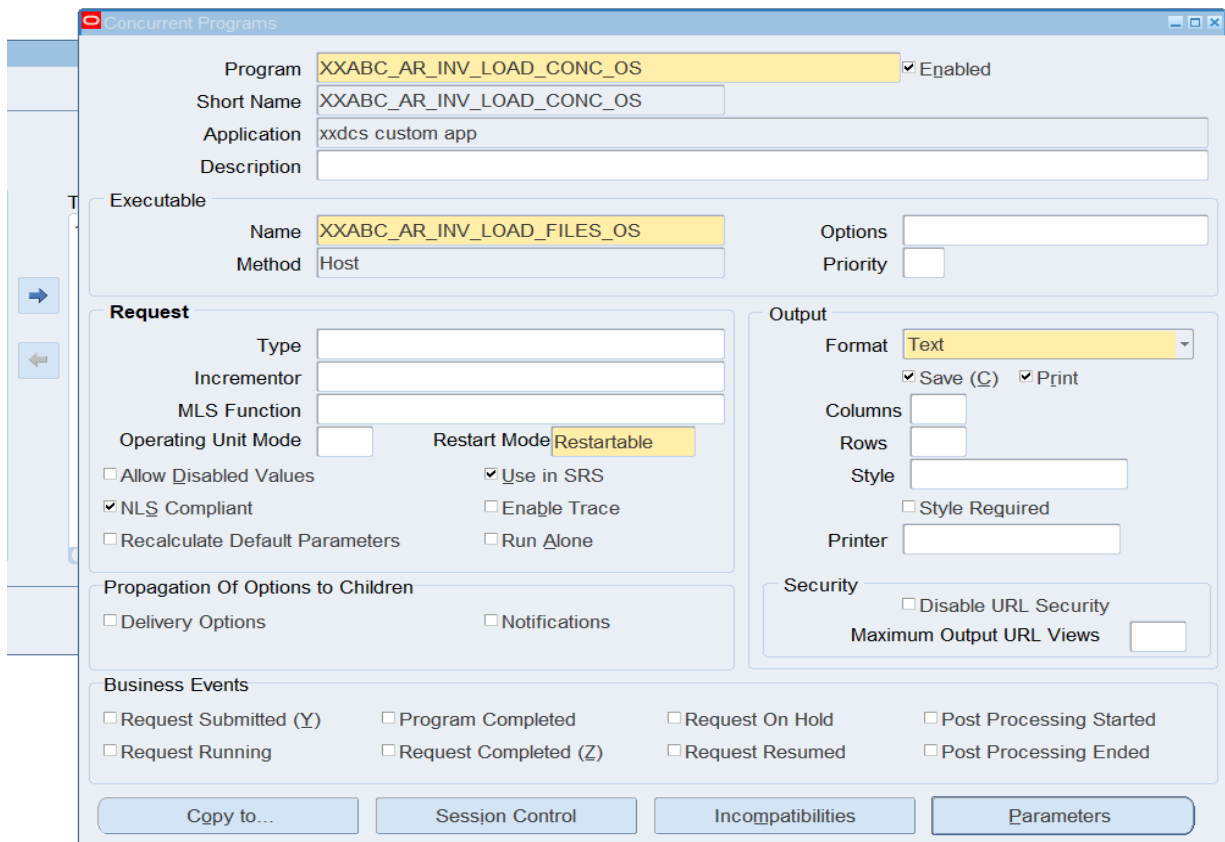
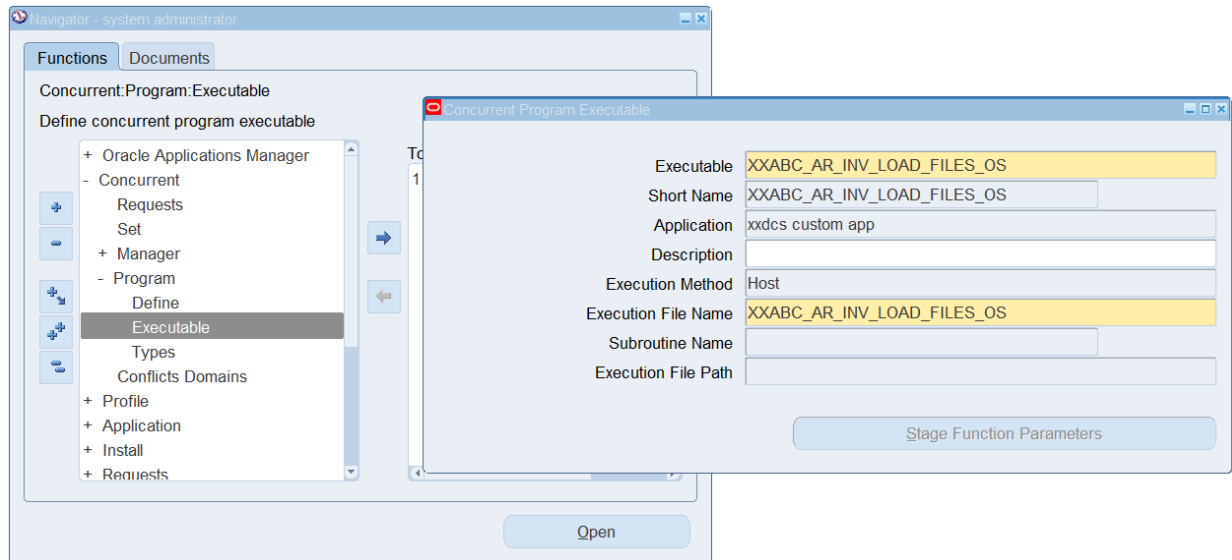
Name	Size	Type	Changed
..		Parent directory	11/13/2024 10:36:56 AM
XMLOffice		File folder	11/7/2024 3:58:05 PM
sql_plsql_questions_DCS		File folder	11/1/2024 9:41:34 PM
petfam_django_project		File folder	7/4/2024 4:20:25 PM
Oracle_app_tech_consultant_notes_DCS		File folder	11/11/2024 11:52:14 PM
office software		File folder	10/31/2024 12:30:51 PM
XXABCARINVINESDATALOAD_ONKAR.ctl	1 KB	CTL File	11/12/2024 10:52:32 PM
XXABCARINVVDATALOAD_ONKAR.ctl	1 KB	CTL File	11/12/2024 10:51:48 PM
XXABC_INV_LINES_DATA_ONKAR.csv	1 KB	Microsoft Excel Co...	11/12/2024 10:50:09 PM
XXABC_INV_HDR_DATA_ONKAR.csv	1 KB	Microsoft Excel Co...	11/12/2024 10:49:45 PM
XXABC_AR_INV_LOAD_FILES_OS.prog	1 KB	PROG File	11/12/2024 10:58:24 PM
ebs_10_0_10_6.ppk	2 KB	PuTTY Private Key ...	10/17/2024 1:10:17 PM


```

Enter command: sed -i 's/\r$/' /u01/install/APPS/fs1/EBApps/appl/xxdcs/12.0.0/bin/
Do not execute commands that require user-input or data transfer
Current directory: /u01/install/APPS/fs1/EBApps/appl/xxdcs/12.0.0/bin/

/u01/install/APPS/fs1/EBApps/appl/xxdcs/12.0.0/bin$ ln -s /u01/in
/u01/install/APPS/fs1/EBApps/appl/xxdcs/12.0.0/bin$ ls -l /u01/in
/u01/install/APPS/fs1/EBApps/appl/xxdcs/12.0.0/bin/XXABC_AR_INV_L
/u01/install/APPS/fs1/EBApps/appl/xxdcs/12.0.0/bin$ chmod +x /u01
/u01/install/APPS/fs1/EBApps/appl/xxdcs/12.0.0/bin$ doc2unix /u01
-bash: line 20: doc2unix: command not found
/u01/install/APPS/fs1/EBApps/appl/xxdcs/12.0.0/bin$ sed -i 's/\r$
  
```

4.REGISTER SHELL SCRIPT AS CONC PROGRAM:



Navigator - system administrator

FunctionsDocuments

Security:User:Define

Define application user

+ Workflow

+ AK

Define Profile Options

- Security

- User

Define

Monitor

- Responsibility

Define

Request

+ ValueSet

+ ORACLE

+ AuditTrail

Web PL/SQL

Top 1.

User NameONKAR

Password

Descriptiontrainee

StatusActive

Person

Customer

Supplier

E-Mail

Fax

Effective Dates

From14-OCT-2024

To

Password Expiration

Days

Accesses

None

Direct Responsibilities

Indirect Responsibilities

Securing Attributes

Responsibility	Application	Description	Security Group	Effective Dates
Accounts Receivables Progr	Receivables		Standard	14-OCT-2024
XML Publisher Administrator	XML Publisher		Standard	14-OCT-2024
HRMS Manager, Progress St	Human Resources		Standard	14-OCT-2024
XXDCS	xxdcs custom app		Standard	08-NOV-2024
Receivables, Vision Operat	Receivables		Standard	11-NOV-2024

Navigator - system administrator

FunctionsDocuments

Security:Responsibility:Define

Define responsibility

- Security

+ User

- Responsibility

Define

Request

+ ValueSet

+ ORACLE

+ AuditTrail

Web PL/SQL

Define Metadata

+ Generic File Manager

+ System Administration

Forms Personalization

Responsibility NameReceivables, Vision Operations (USA)

ApplicationReceivables

Responsibility KeyRECEIVABLES_VISION_OPERATIONS

DescriptionOracle Receivables Superuser, Vision O

Effective Dates

From07-DEC-1997

To

Available From

Oracle Applications

Oracle Self Service Web Applications

Oracle Mobile Applications

Data Group

NameStandard

ApplicationReceivables

Request Group

NameReceivables All

ApplicationReceivables

MenuAR_NAVIGATE_GUI

Web Host Name

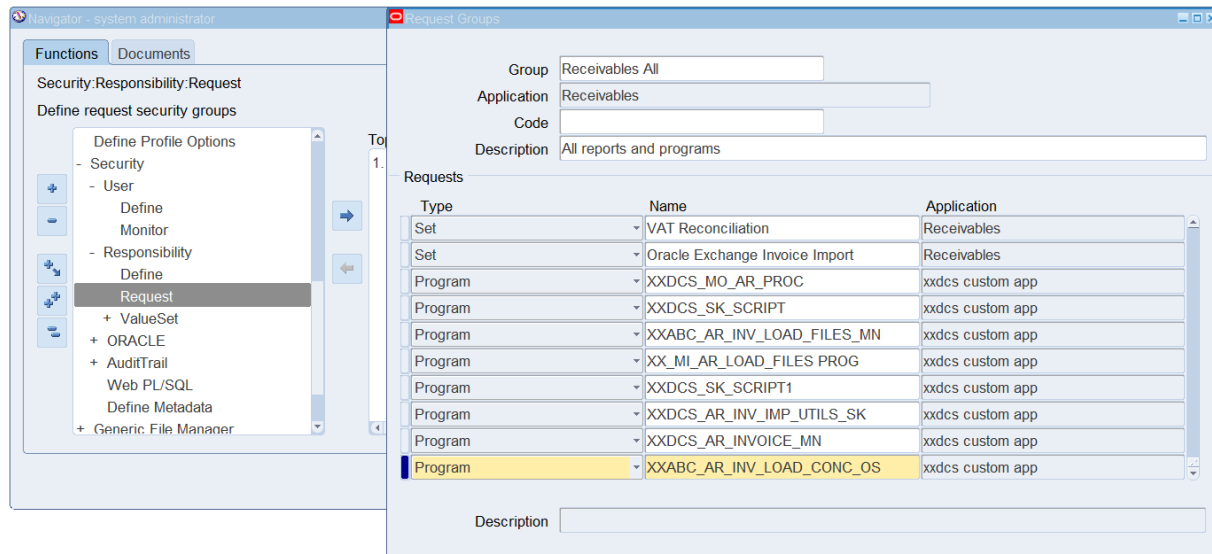
Web Agent Name

Menu Exclusions

Excluded Items

Securing Attributes

Type	Name	Description
Function		



RUN THE REQUEST FROM RECEIVABLE RES

```
+-----+
xxdcs custom app: Version : 12.2

Copyright (c) 1998, 2013, Oracle and/or its affiliates. All rights reserved.

XXABC_AR_INV_LOAD_CONC_OS: XXABC_AR_INV_LOAD_CONC_OS
+-----+

Current system time is 13-NOV-2024 06:24:50

+-----+

=====
Entered Program
Invoking Header Load

SQL*Loader: Release 10.1.0.5.0 - Production on Wed Nov 13 06:24:50 2024

Copyright (c) 1982, 2005, Oracle. All rights reserved.

Commit point reached - logical record count 2
Commit point reached - logical record count 3
Invoking Lines Load

SQL*Loader: Release 10.1.0.5.0 - Production on Wed Nov 13 06:24:50 2024

Copyright (c) 1982, 2005, Oracle. All rights reserved.

Commit point reached - logical record count 6
Commit point reached - logical record count 7
End of script

+-----+
No completion options were requested.

Output file size:

+-----+
Concurrent request completed successfully
Current system time is 13-NOV-2024 06:24:50
+-----+
```

**NOW STAGING TABLE GETS LOADED.(XXABC_AR_INV_HDR_STG_ONKAR,
XXABC_AR_INV_LINES_STG_ONKAR)**

<pre> SELECT * FROM XXABC_AR_INV_HDR_STG_ONKAR; SELECT * FROM XXABC_AR_INV_LINES_STG_ONKAR; </pre>												
Query Result * Query Result 1 *												
All Rows Fetched: 2 in 0.009 seconds												
TRX_HEADER_ID	TRX_NUMBER	TRX_DATE	TRX_CURRENCY	CUST_TRX_TYPE_ID	BILL_TO_CUSTOMER_ID	TERM_ID	FINANCE_CHARGES	STATUS_TRX	PRINTING_OPTION	RECORD_NUM	STATUS	ERROR_MSG
1	1 (null)	13-NOV-24	USD	1563	3347	1062 N	OP	NOT		2 I	(null)	
2	2 (null)	13-NOV-24	USD	1563	3347	1062 N	OP	NOT		3 I	(null)	
<pre> SELECT * FROM XXABC_AR_INV_HDR_STG_ONKAR; SELECT * FROM XXABC_AR_INV_LINES_STG_ONKAR; </pre>												
Query Result * Query Result 1 *												
All Rows Fetched: 6 in 0.014 seconds												
TRX_HEADER_ID	TRX_LINE_ID	LINE_NUMBER	INVENTORY_ITEM_ID	DESCRIPTION	QUANTITY_INVOICED	UNIT_SELLING_PRICE	UOM_CODE	LINE_TYPE	RECORD_NUM	STATUS	ERROR_MSG	
1	1	1	1	(null) Header ID1-Line1	5	2500 (null)	LINE	(null) I	(null)	(null)		
2	1	2	2	(null) Header ID1-Line2	9	3500 (null)	LINE	(null) I	(null)	(null)		
3	1	3	3	(null) Header ID1-Line1	5	4500 (null)	LINE	(null) I	(null)	(null)		
4	2	1	1	(null) Header ID1-Line1	9	5500 (null)	LINE	(null) I	(null)	(null)		
5	2	2	2	(null) Header ID1-Line2	5	6500 (null)	LINE	(null) I	(null)	(null)		
6	2	3	3	(null) Header ID1-Line3	9	7500 (null)	LINE	(null) I	(null)	(null)		

5. INVOKE THE PUBLIC API:

WE HAVE REGISTERED CUSTOM PLSQL API AS CONC PROG WHICH WILL CALL THE PUBLIC API. IT CONTAINS ‘IMPORT_DATA’ METHOD WHICH CALLS THE ‘VALIDATE_DATA’ METHOD WHICH VALIDATE STAGING TABLE DATA AND IF THEIR IS ERROR THEN IT RETURN ‘RETCODE’(ERROR CODE) i.e. errbuf (to hold error messages) and retcode (to hold status codes, e.g., 0 for success or non-zero for failure).

The screenshot shows the SQL Developer interface with a code editor. The package body is being created for the package XXABC_AR_INV_IMP_UTILS. The code includes a procedure VALIDATE_DATA that performs validation on staging data for AR invoices. It uses two cursors: C_HDR for header data and C_LINES for line data. The procedure includes loops and conditional logic to update the status and error message for invalid data.

```
CREATE OR REPLACE PACKAGE BODY XXABC_AR_INV_IMP_UTILS
IS
--Method to validate staging data for AR Invoice
PROCEDURE VALIDATE_DATA
IS
    CURSOR C_HDR IS SELECT * FROM XXABC_AR_INV_HDR_STG_ONKAR ;
    CURSOR C_LINES IS SELECT * FROM XXABC_AR_INV_LINES_STG_ONKAR ;
BEGIN
    --Validation for Header Data
    FOR REC IN C_HDR
    LOOP

        IF rec.trx_currency is null THEN
            UPDATE XXABC_AR_INV_HDR_STG_ONKAR
            SET STATUS='E' , ERROR_MSG = ERROR_MSG || ' Invalid Curr'
            WHERE trx_header_id=rec.trx_header_id;
        END IF ;

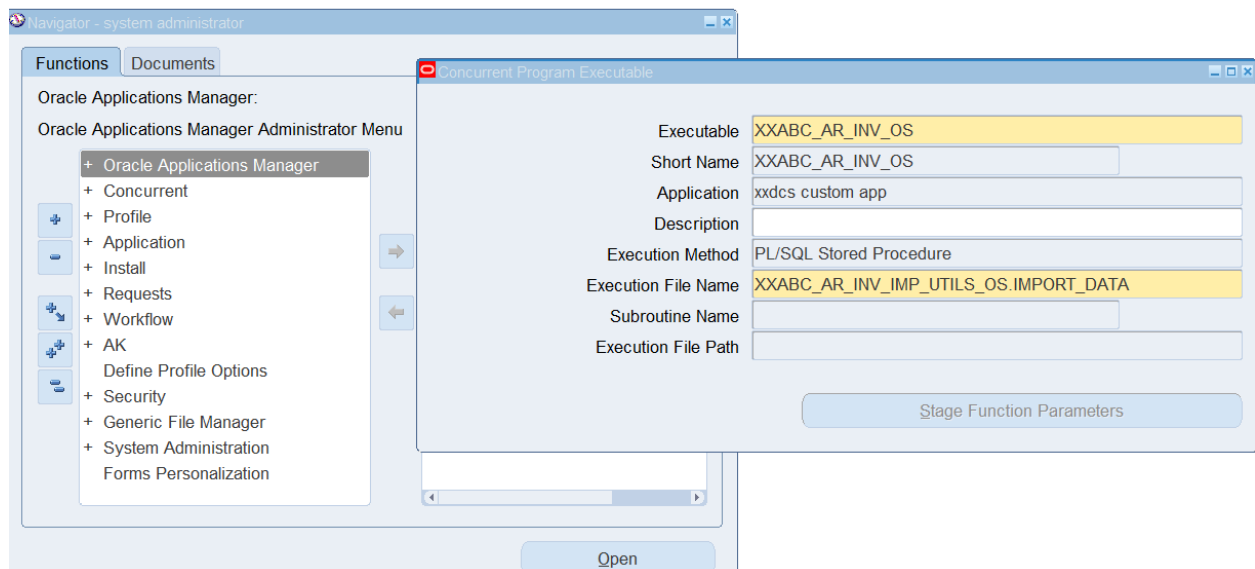
        IF rec.term_id is null THEN
            UPDATE XXABC_AR_INV_HDR_STG_ONKAR
            SET STATUS='E' , ERROR_MSG = ERROR_MSG || ' Invalid Pay Terms'
            WHERE trx_header_id=rec.trx_header_id;
        END IF ;

    END LOOP;

    --Validation for Lines Data
    FOR REC IN C_LINES
    LOOP
        NULL ;
    END LOOP;

END VALIDATE_DATA;
```

NOW CREATE EXE, CONC PRO. AND REGISTER THE METHOD:



NOW CREATE CONC PRO > ATTACH WITH REQ. GRP OF RECEIVABLES, VISION OPE. >
AND RUN THE REQUEST FROM RECEIVABLES RES.

The image shows two screenshots of the Oracle EBS system administrator interface. The top screenshot displays the 'Concurrent Programs' window, and the bottom screenshot displays the 'Request Groups' window.

Concurrent Programs Window:

- Program:** XXABC_AR_INV_CONC_OS (Enabled)
- Short Name:** XXABC_AR_INV_CONC_OS
- Application:** xxdcs custom app
- Description:**
- Executable:**
 - Name:** XXABC_AR_INV_OS
 - Method:** PL/SQL Stored Procedure
 - Options:**
 - Priority:**
- Request:**
 - Type:**
 - Incrementor:**
 - MLS Function:**
 - Operating Unit Mode:**
 - Restart Mode:** Restartable
 - ☐ Allow Disabled Values
 - ☒ NLS Compliant
 - ☐ Recalculate Default Parameters
 - ☒ Use in SRS
 - ☐ Enable Trace
 - ☐ Run Alone
- Output:**
 - Format:** Text
 - ☒ Save (C) ☒ Print
 - Columns:**
 - Rows:**
 - Style:**
 - ☐ Style Required
 - Printer:**
- Propagation Of Options to Children:**
 - ☐ Delivery Options
 - ☐ Notifications
- Security:**
 - ☐ Disable URL Security
 - Maximum Output URL Views:**

Request Groups Window:

- Group:** Receivables All
- Application:** Receivables
- Code:**
- Description:** All reports and programs
- Requests:**

Type	Name	Application
Program	XX_AY_AR_INV_IMPORT_DATA	xxdcs custom app
Program	XXDCS_AR_INV_IMP_UTILS_PKB_HH	xxdcs custom app
Program	XXDCS_SK_SCRIPT1	xxdcs custom app
Program	XXDCS_AR_INV_IMP_UTILS_SK	xxdcs custom app
Program	XXDCS_AR_INVOICE_MN	xxdcs custom app
Program	XXDCS_MO_AR_PROC	xxdcs custom app
Program	XXABC_AR_INV_CONC_OS	xxdcs custom app
Program	XX_AY_1_AR_INV_CREATE	xxdcs custom app
Set	VAT Reconciliation	Receivables
Set	Oracle Exchange Invoice Import	Receivables
- Description:**

IN VIEW LOG WE HAVE CREATED TWO INVOICE NO.

```
+-----+
xxdcs custom app: Version : 12.2
```

```
Copyright (c) 1998, 2013, Oracle and/or its affiliates. All rights reserved.
```

```
XXABC_AR_INV_CONC_OS: XXABC_AR_INV_CONC_OS
+-----+
```

```
Current system time is 13-NOV-2024 10:20:19
```

```
+-----+
[13-NOV-2024 10:20:19]: **Starts**
```

```
[13-NOV-2024 10:20:20]: **Ends**
+-----+
```

```
Start of log messages from FND_FILE
+-----+
```

```
Created:S
```

```
Customer Trx Number 12067
```

```
ORA-0000: normal, successful completion
```

```
Created:S
```

```
Customer Trx Number 12068
```

```
ORA-0000: normal, successful completion
+-----+
```

```
End of log messages from FND_FILE
+-----+
```

```
+-----+
No completion options were requested.
```

```
Output file size:
```

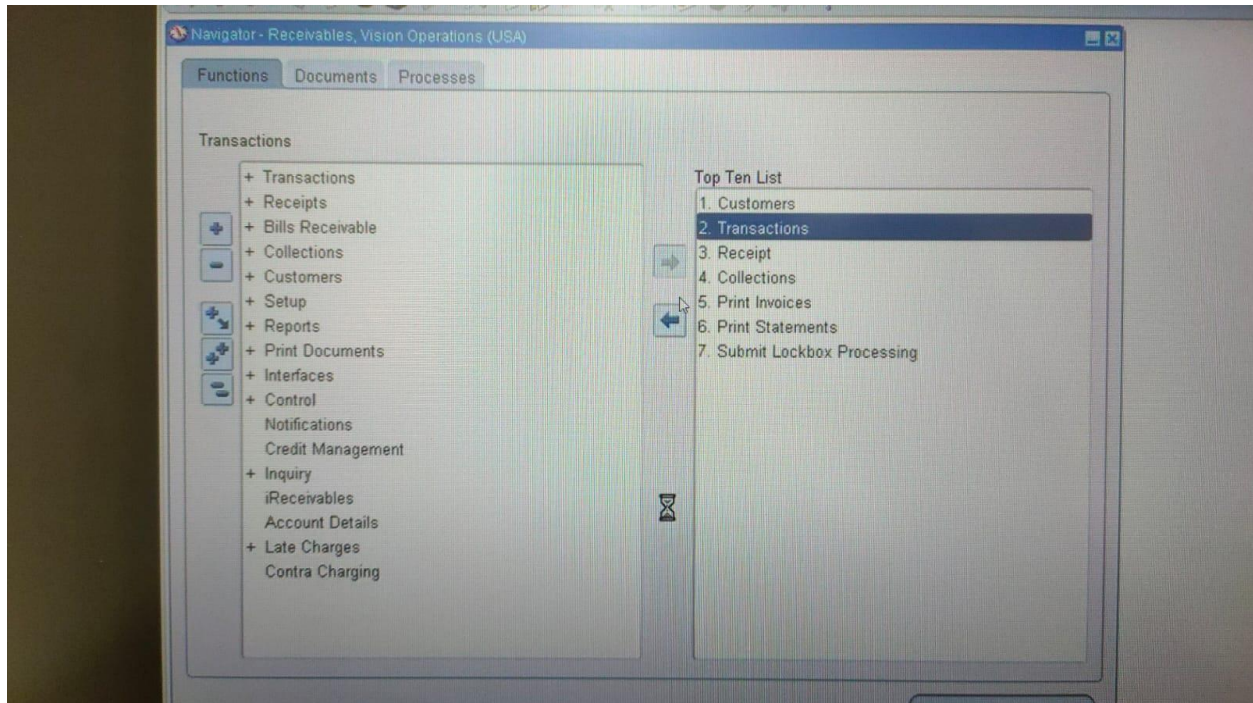
```
0
```

```
Deleting empty output file.
```

```
+-----+
Concurrent request completed successfully
```

```
Current system time is 13-NOV-2024 10:20:20
+-----+
```

**NOW CHECK THE INVOICE NO :
RECEIVABLES, VISION ... > TRANSACTION>TRANSACTION**



transactions

Transaction

Source Date

Number 12067 GL Date

Class Currency

Type Document Num

Reference Transaction []

Legal Entity ☐ Complete

Balance Due

Line

Tax

Freight

Charges

Total

Main More Notes Commitment Reference Information

Ship To

Name

Number

Location

Address

Contact

Bill To

Name

Number

Location

Address

Contact

Sold To

Name

Number

Paying Customer

Name

Number

Location

Payment Details

Receipt Method

Payment Method

Instrument Number

Commitment

Payment Term

Invoicing Rule

Due Date

Transactions (Vision Operations - USD)

Transaction

Source: Manual
 Number: 12067
 Class: Invoice
 Type: Invoice-OKS
 Reference:
 Legal Entity: Vision Operations

Date: 13-NOV-2024
 GL Date: 31-DEC-2016
 Currency: USD
 Document Num:
 Transaction: []

Balance Due

Line	66,500.00
Tax	0.00
Freight	0.00
Charges	0.00
Total	66,500.00

Details Refresh

Main More Notes Commitment Reference Information

Ship To

Name:
 Number:
 Location:
 Address:
 Contact:
 Commitment:
 Payment Term: N30
 Invoicing Rule:
 Due Date: 13-DEC-2024

Bill To

Business World
 1608
 San Jose (OPS)
 2391 L Street
 San Jose, CA 95106 United States

Sold To

Name: Business World
 Number: 1608

Paying Customer

Name: Business World
 Number: 1608
 Location: San Jose (OPS)

Payment Details

Receipt Method: Automatic Receipts
 Payment Method: Bank Account Transfer
 Instrument Number: XXXXXXXXXX0012
 Select Instrument

Line Items Tax Freight Distributions Sales Credits Incomplete

CLICK LINE ITEMS:

Transactions (Vision Operations - USD)

Lines (Vision Operations: USD) - Business World, 12067

Transaction: 66,500.00
 Lines: 66,500.00
 Tax: 0.00
 Freight: 0.00

Main Sales Order Tax Exemption Rules Ship To Information More

Num	Item	Description	UOM	Quantity	Unit Price	Amount	Tax Classification	Trans
1		Header ID1-Line1		5	2500	12,500.00		-
2		Header ID1-Line2		9	3500	31,500.00		-
3		Header ID1-Line1		5	4500	22,500.00		-

NOW OUT DATA IS LOADED PERFECTLY IN BASE TABLE I.E. **ra_customer_trx_all**,
ra_customer_trx_lines_all

PLSQL TO CALL API:

create or replace package BODY XXABC_AR_INV_IMP_UTILS_OS

IS

--Method to validate staging data for AR Invoice

PROCEDURE VALIDATE_DATA

IS

CURSOR C_HDR IS SELECT * FROM XXABC_AR_INV_HDR_STG_ONKAR ;

CURSOR C_LINES IS SELECT * FROM XXABC_AR_INV_LINES_STG_ONKAR ;

BEGIN

--Validation for Header Data

FOR REC IN C_HDR

LOOP

IF rec.trx_currency is null THEN

UPDATE XXABC_AR_INV_HDR_STG_ONKAR

SET STATUS='E' , ERROR_MSG = ERROR_MSG ||' Invalid Curr'

WHERE trx_header_id=rec.trx_header_id;

END IF ;

IF rec.term_id is null THEN

UPDATE XXABC_AR_INV_HDR_STG_ONKAR

SET STATUS='E' , ERROR_MSG = ERROR_MSG ||' Invalid Pay Terms'

WHERE trx_header_id=rec.trx_header_id;

END IF ;

END LOOP;

--Validation for Lines Data


```
FOR REC IN C_LINES
```

```
LOOP
```

```
NULL ;
```

```
END LOOP;
```

```
END VALIDATE_DATA;
```

```
--Method to invoke public API
```

```
PROCEDURE IMPORT_DATA(ERRBUF OUT VARCHAR2, RETCODE OUT VARCHAR2)
```

```
IS
```

```
L_HDR_ERR_CNT NUMBER;
```

```
L_LINES_ERR_CNT NUMBER ;
```

```
l_return_status    varchar2(1);
```

```
l_msg_count        number;
```

```
l_msg_data         varchar2(2000);
```

```
l_batch_source_rec ar_invoice_api_pub.batch_source_rec_type; -- ar_invoice_api_pub  
is package & batch_source_rec_type is record type
```

```
l_trx_header_tbl   ar_invoice_api_pub.trx_header_tbl_type; -- parameter of api is itself a  
table/rec type which contains multiple attributes in it.
```

```
l_trx_lines_tbl     ar_invoice_api_pub.trx_line_tbl_type; -- it will contain lines data which  
will be loaded from our lines staging table
```

```
l_trx_dist_tbl      ar_invoice_api_pub.trx_dist_tbl_type; -- we are not using this in this e.g.
```

```
l_trx_salescredits_tbl ar_invoice_api_pub.trx_salescredits_tbl_type; -- we are not using  
this in this e.g.
```

```
l_cust_trx_id       number;
```

```
l_customer_trx_id number;
```

```
l_cnt number;
```

```
l_err_msg varchar2(1000);
```

```
l_inv_number VARCHAR2(50);
```

```
l_line_cnt number;
```

```
l_org_id number;
```

```
BEGIN
```

```
--MO_GLOBAL.SET_POLICY_CONTEXT('S',204);
```

```
--Call the validate data to validate staging table data
```

```
VALIDATE_DATA ;
```

```
SELECT COUNT(*) INTO L_HDR_ERR_CNT
```

```
FROM XXABC_AR_INV_HDR_STG_ONKAR
```

```
WHERE STATUS='E';
```

```
SELECT COUNT(*) INTO L_LINES_ERR_CNT
```

```
FROM XXABC_AR_INV_LINES_STG_ONKAR
```

```
WHERE STATUS='E';
```

```
IF( L_HDR_ERR_CNT >0 or L_LINES_ERR_CNT >0 ) THEN
```

```
FND_FILE.PUT_LINE(FND_FILE.LOG,'Issue with staging table data,
```

```
Plese recheck and reload');
```

```
retcode := 2;
```

```

        errbuf:= 'Data Issue Please Recheck';

        commit;

    END IF ;

--Populate Table Type Values

--header data

l_cnt := 1 ;

for rec in (select * from XXABC_AR_INV_HDR_STG_ONKAR) --implicit cursor / it will stop
itself as data gets empty.

    LOOP

        l_batch_source_rec.batch_source_id := -1;

        l_trx_header_tbl(1).trx_header_id := rec.trx_header_id; -- we are creating
collection(nested table) to store the values.

        l_trx_header_tbl(1).trx_date      := sysdate;

        l_trx_header_tbl(1).trx_currency := rec.trx_currency;

        l_trx_header_tbl(1).cust_trx_type_id := rec.cust_trx_type_id;

        l_trx_header_tbl(1).bill_to_customer_id := rec.bill_to_customer_id;

        l_trx_header_tbl(1).term_id      := rec.term_id;

        l_trx_header_tbl(1).finance_charges := rec.finance_charges;

        l_trx_header_tbl(1).status_trx := rec.status_trx;

        l_trx_header_tbl(1).printing_option := rec.printing_option;

l_cnt:=l_cnt+1 ;

--lines data

l_line_cnt:=1;

```

```
for rec_lines in ( select * from XXABC_AR_INV_LINES_STG_ONKAR lines
WHERE lines.trx_header_id=rec.trx_header_id) --implicit cursor / it will stop itself as
data gets empty.
```

LOOP

```
l_trx_lines_tbl(l_line_cnt).trx_header_id := rec_lines.trx_header_id;
l_trx_lines_tbl(l_line_cnt).trx_line_id := rec_lines.trx_line_id;
l_trx_lines_tbl(l_line_cnt).line_number := rec_lines.line_number;
l_trx_lines_tbl(l_line_cnt).inventory_item_id := NULL;
l_trx_lines_tbl(l_line_cnt).description := rec_lines.description;
l_trx_lines_tbl(l_line_cnt).quantity_invoiced := rec_lines.quantity_invoiced;
l_trx_lines_tbl(l_line_cnt).unit_selling_price := rec_lines.unit_selling_price;
l_trx_lines_tbl(l_line_cnt).uom_code := NULL;
l_trx_lines_tbl(l_line_cnt).line_type := rec_lines.line_type;
l_line_cnt:=l_line_cnt+1;
```

END LOOP;

--Invoking Public API

--Here we call the API to create Invoice with the stored values

AR_INVOICE_API_PUB.create_single_invoice(-- WE ARE PROVIDING I/P AND GETTING
O/P FROM API SIMOULTANEOUSLY

p_api_version =>1.0, -- GIVING I/P TO API

p_init_msg_list =>NULL,

p_commit =>NULL,

```

p_batch_source_rec    => l_batch_source_rec,
p_trx_header_tbl      => l_trx_header_tbl,
p_trx_lines_tbl       => l_trx_lines_tbl,
p_trx_dist_tbl        => l_trx_dist_tbl,
p_trx_salescredits_tbl => l_trx_salescredits_tbl,
x_customer_trx_id     => l_customer_trx_id, -- TAKING O/P FROM API
x_return_status       => l_return_status,
x_msg_count           => l_msg_count,
x_msg_data            => l_msg_data
);

```

fnd_file.put_line(fnd_file.log,'Created:'||l_msg_data||l_return_status); -- fnd_file.put_line
to write data to a log file

-- fnd_file.log package that represents a log file that
you can use to write messages

SELECT count(*) Into l_cnt From ar_trx_errors_gt; -- checks if there is no error log in
ar_trx_errors_gt table then proceed further

IF l_cnt = 0 THEN

BEGIN

SELECT trx_number INTO l_inv_number -- getting value from base table & generating
trx no of customer in our log file.

FROM ra_customer_trx_all

WHERE customer_trx_id = l_customer_trx_id; -- TAKING O/P FROM API

EXCEPTION

WHEN OTHERS THEN

NULL;

```

END;

END IF ;

fnd_file.put_line(fnd_file.log, 'Customer Trx NUmber ' || l_inv_number);

IF l_return_status = fnd_api.g_ret_sts_error OR
l_return_status = fnd_api.g_ret_sts_unexp_error THEN

fnd_file.put_line(fnd_file.log,':'||l_return_status||':'||sqlerrm);

Else

dbms_output.put_line(''||l_return_status||':'||sqlerrm);

If (ar_invoice_api_pub.g_api_outputs.batch_id IS NOT NULL) Then

fnd_file.put_line(fnd_file.log,'Invoice(s) suceessfully created!') ;

fnd_file.put_line(fnd_file.log,'Batch ID: ' ||
ar_invoice_api_pub.g_api_outputs.batch_id);

fnd_file.put_line(fnd_file.log,'customer_trx_id: ' || l_cust_trx_id);

Else

fnd_file.put_line(fnd_file.log,''||sqlerrm);

End If;

end if;

--Clear the data fom record

l_trx_lines_tbl.delete ;

END LOOP;

END IMPORT_DATA;

END XXABC_AR_INV_IMP_UTILS_OS;

```