

Creation of Item

Step 1: Create a Staging table.

```
create table Item_staging_tbl(  segment1 varchar2(40),
                                description varchar2(150),
                                template_name varchar2(40),
                                organization_code varchar2(3),
                                process_flag char(1),
                                error_message varchar2(2000),
                                creation_date date,
                                created_by number,
                                last_update_date date,
                                last_updated_by number);
```

Step 2 : Create a CTL file.

```
options(SKIP=0)
LOAD DATA
infile *
TRUNCATE
INTO TABLE Item_staging_tbl
FIELDS TERMINATED BY ','
OPTIONALLY ENCLOSED BY '"'
TRAILING NULLCOLS ( segment1,
                     description,
                     template_name,
                     organization_code,
                     process_flag      "nvl(:process_flag, 'N')",
                     error_message,
                     creation_date     SYSDATE,
                     created_by        "fnd_global.user_id",
                     last_update_date  SYSDATE,
                     last_updated_by   "fnd_global.user_id")
```

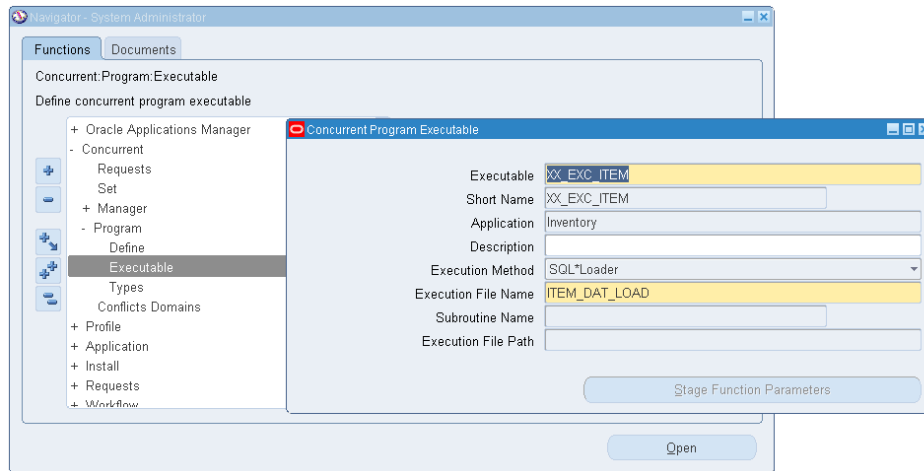
Step 3: Create a CSV File.

909090	Tower	Purchased Item	V1	N
909091	Leg	Purchased Item	V1	N
909092	Bracing	Purchased Item	V1	N
909093	Stub	Purchased Item	V1	N

Step 4: Create a Concurrent Program to Load the Data from CSV file to Staging table.

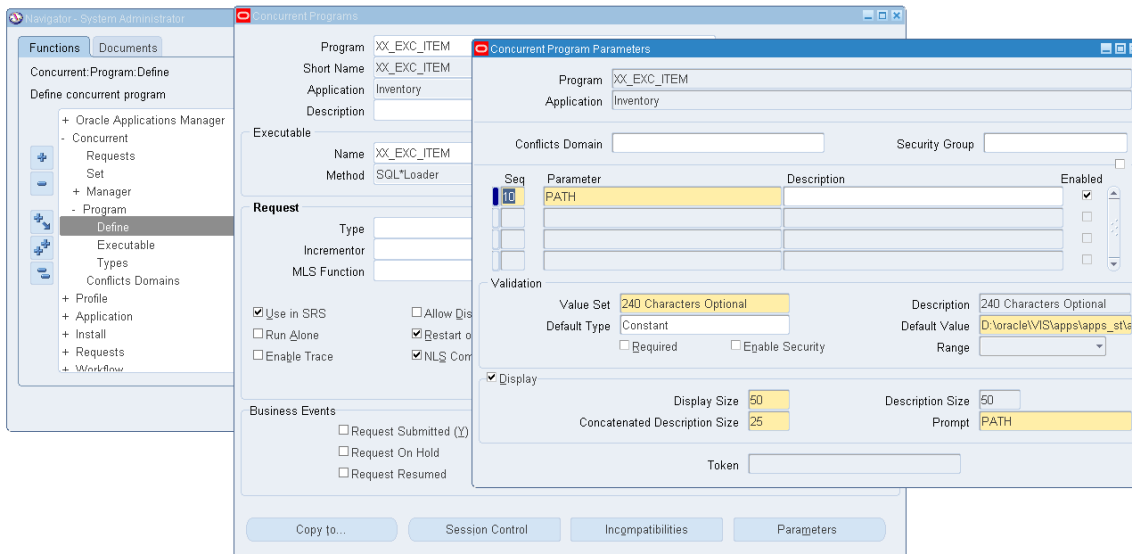
➤ Create Executable

- Executable : **XX_EXC_ITEM**
- Short Name: **XX_EXC_ITEM**
- Application: **Inventory**
- Execution Method: **SQL*Loader**
- Execution File Name: **ITEM_DAT_LOAD**



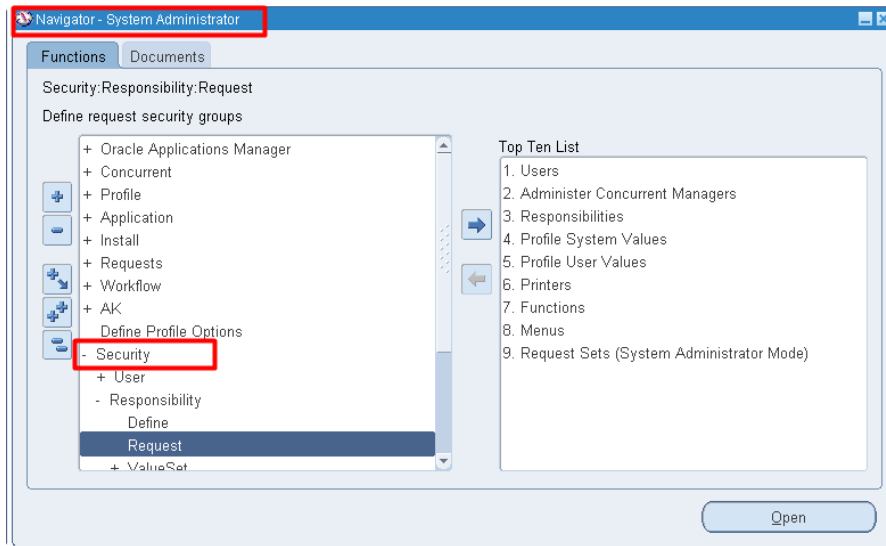
➤ Create Concurrent Program for that executable.

- Program: **XX_EXC_ITEM**
- Short Name: **XX_EXC_ITEM**
- Application: **Inventory**
- Executable Name : **XX_EXC_ITEM**
- Method: **SQL*Loader**
- Parameters
 - Path: D:\oracle\VIS\apps\apps_st\appl\inv\12.0.0\bin\XXCUSITEMIMPORT.csv



➤ Request Group

Navigation → System Administrator → Security → Request



Responsibility name- Inventory, Vision Operation(USA)

Application – Inventory

Request Group Name – All Inclusive GUI

The screenshot shows the 'Responsibilities' window. The 'Responsibility Name' is 'Anitha inventory superuser'. The 'Application' is 'Inventory'. The 'Responsibility Key' is 'ANITHA_INV'. The 'Effective Dates' are 'From 04-FEB-2016' to 'To'. The 'Available From' section has 'Oracle Applications' selected. The 'Data Group' section has 'Name Standard' and 'Application Inventory'. The 'Request Group' section has 'Name All Inclusive GUI' and 'Application Inventory'. The 'Menu' is 'INV_NAVIGATE'. The 'Web Host Name' and 'Web Agent Name' are empty. The 'Menu Exclusions' tab is active, showing a table with columns 'Type', 'Name', and 'Description'.

Type	Name	Description
Function		

➤ **Assign to Suitable Responsibility**

Assign the Concurrent Program to the existing Group

The 'Request Groups' window is shown with the following fields:

- Group: All Inclusive GUI
- Application: Inventory
- Code: INV_ALL_REPORTS_GUI
- Description: All inclusive reports GUI

The 'Requests' section contains a table with the following data:

Type	Name	Application
Program	Inventory Packing Slip	Inventory
Program	XX_EXC_ITEM	Inventory
Program	XX_ITEM_INTERFACE	Inventory
Program	XX_ITEM_INTERFACE1	Inventory
Program	Overhead Report	Bills of Material
Program	Elemental Inventory Value Report - by Sub	Bills of Material
Program	Inventory Value Report - by Subinventory	Bills of Material
Program	All Inventories Value Report	Bills of Material
Program	Intransit Value Report	Bills of Material
Program	Subinventory Account Value Report	Bills of Material

At the bottom, there is a 'Description' field.

Go to Inventory, Vision Operation(USA) Responsibility.

View → Request → submit a new Request

- Name → **XX_EXC_ITEM**
- Parameters

The 'Submit Request' window is shown with the following fields:

- Run this Request...: Copy...
- Name: XX_EXC_ITEM
- Operating Unit:
- Parameters:
- Language: American English
- At these Times...: Run the...
- Upon Completion: Layout, Notify, Print to: noprint, Options...
- Buttons: Help (C), Submit, Cancel

The 'Parameters' window is shown with the following fields:

- PATH: D:\ORACLE\VIS\APPS\APPS_ST\APPL\INV\12.0.0\BIN\XXCUSITEMIMPORT.CSV
- Buttons: OK, Cancel, Clear, Help

Request ID	Name	Parent	Phase	Status	Parameters
5947794	To submit item import from		Completed	Warning	, 1, 1, 1, 1, 1
5947793	XX_ITEM_INTERFACE		Completed	Normal	
5947792	XX_EXC_ITEM		Completed	Normal	D:\ORACLE\VIS\APPS\APPS
5947791	XX_EXC_ITEM		Completed	Error	D:\ORACLE\VIS\APPS\APPS
5947789	XX_ITEM_INTERFACE		Completed	Normal	
5947784	XX_ITEM_INTERFACE1		Completed	Normal	D:\ORACLE\VIS\APPS\APPS
5947783	XX_ITEM_INTERFACE1		Completed	Normal	D:\ORACLE\VIS\APPS\APPS
5947782	XX_ITEM_INTERFACE1		Completed	Normal	D:\ORACLE\VIS\APPS\APPS
5947780	XX_ITEM_INTERFACE1		Completed	Normal	D:\ORACLE\VIS\APPS\APPS
5947779	XX_ITEM_INTERFACE1		Completed	Warning	D:\ORACLE\VIS\APPS\APPS

Then all the data in CSV file was load in Staging table

```

29
30 SELECT * FROM item_staging_tbl;

```

SEGMENT1	DESCRIPTION	TEMPLATE_NAME	ORGANIZATION_CODE	PROCESS_FLAG	ERROR_MESSAGE	CREATION_DATE	CREATED_BY	LAST_UPDATE_DATE	LAST_UPDATED_BY
1 909090	Tower	Purchased Item V1		N	(null)	06-FEB-24	-106-FEB-24		-1
2 909091	Leg	Purchased Item V1		N	(null)	06-FEB-24	-106-FEB-24		-1
3 909092	Bracing	Purchased Item V1		N	(null)	06-FEB-24	-106-FEB-24		-1
4 909093	Stub	Purchased Item V1		N	(null)	06-FEB-24	-106-FEB-24		-1

Then Create a Package with All required Validation .

The data will import from Stagging table to Interface table and Also Interface table to base table ..

Package:

```
CREATE OR REPLACE PACKAGE pkg_item_import1
```

```
IS
```

```
    PROCEDURE prc_item_import1(errbuf OUT VARCHAR2, retcode OUT NUMBER);
```

```
END;
```

```
CREATE OR REPLACE PACKAGE BODY pkg_item_import1
```

```
IS
```

```
    g_user_id fnd_user.user_id%TYPE;
```

```
    g_login_id NUMBER(15) := 0;
```

```
    PROCEDURE prc_item_import1(errbuf OUT VARCHAR2, retcode OUT NUMBER)
```

```
    IS
```

CURSOR c1

IS

```
SELECT a.ROWID row_id, a.*
FROM item_staging_tbla
WHERE a.process_flag = 'N';
```

e_flag CHAR(1);

e_msg VARCHAR2(2000);

l_count NUMBER := 0;

n_organization_id NUMBER;

xx_org VARCHAR2(10);

n_segment1 VARCHAR2(100);

lv_template_name varchar2(100);

n_request_id number;

BEGIN

DBMS_OUTPUT.put_line ('ENTER THE LOOP');

FOR i IN c1

LOOP

e_flag := 'Y';

l_count := l_count + 1;

e_msg := NULL;

DBMS_OUTPUT.put_line ('1. ' || i.organization_code);

SELECT organization_id

INTO n_organization_id

FROM mtl_parameters

WHERE organization_code = i.organization_code;

----- organization_code VALIDATION -----

IF i.organization_code IS NOT NULL

THEN

BEGIN

SELECT organization_code

INTO xx_org

FROM mtl_parameters

WHERE organization_id =

(SELECT master_organization_id

FROM mtl_parameters

WHERE organization_code = i.organization_code);

DBMS_OUTPUT.put_line ('THE CODE ORG CODE IS ' || xx_org);

EXCEPTION

```

        WHEN OTHERS
        THEN
            e_flag := 'E';
            e_msg := 'organization code IS INVLD';
            DBMS_OUTPUT.put_line ('CODE IS ' || e_msg);
        END;
    END IF;

```

----- ITEM CODE VALIDATIONS -----

```

    IF i.segment1 IS NOT NULL
    THEN
        BEGIN
            SELECT 1
            INTO n_segment1
            FROM mtl_system_items_b msib
            WHERE segment1 = i.segment1
            AND organization_id = n_organization_id;

            IF n_segment1 <> 0
            THEN
                e_msg := 'Item Code is Existed';
                DBMS_OUTPUT.put_line ('N_SEGMENT1: ' || l_count);
            END IF;
        EXCEPTION
        WHEN OTHERS
        THEN
            NULL;
        END;
    END IF;

```

----- Template Name is Existed or not -----

```

    IF i.template_name IS NOT NULL
    THEN
        BEGIN
            SELECT template_name
            INTO lv_template_name
            FROM mtl_item_templates
            WHERE template_name = i.template_name;
        EXCEPTION
        WHEN OTHERS
        THEN
            e_msg := e_msg || ', ' || 'Template name is not existed';
            DBMS_OUTPUT.put_line ('n_template: ' || l_count);
        END;
    END IF;

```

```

END;
END IF;

IF e_flag = 'Y'
THEN
    INSERT INTO mtl_system_items_interface
        (segment1, description, template_name,
         organization_code,
         organization_id,
         transaction_type, process_flag, creation_date,
         created_by, last_update_date, last_updated_by
        )
    VALUES (i.segment1, i.description, i.template_name,
            i.organization_code,
            (SELECT organization_id
             FROM mtl_parameters
             WHERE organization_code = i.organization_code),
            'CREATE', 1, SYSDATE,
            fnd_global.user_id, SYSDATE, fnd_global.user_id
            );

    IF SQL%ROWCOUNT <> 0
    THEN
        UPDATE item_staging_tbl
            SET process_flag = 'S'
            WHERE ROWID = i.row_id;

        COMMIT;
    END IF;
ELSE
    UPDATE item_staging_tbl
        SET process_flag = 'E'
        WHERE ROWID = i.row_id;
    END IF;
END LOOP;
n_request_id :=
fnd_request.submit_request('INV',
                           'INCOIN',
                           'To submit item import from backend',
                           NULL,
                           FALSE,
                           fnd_profile.VALUE('MFG_ORGANIZATION_ID'),

```



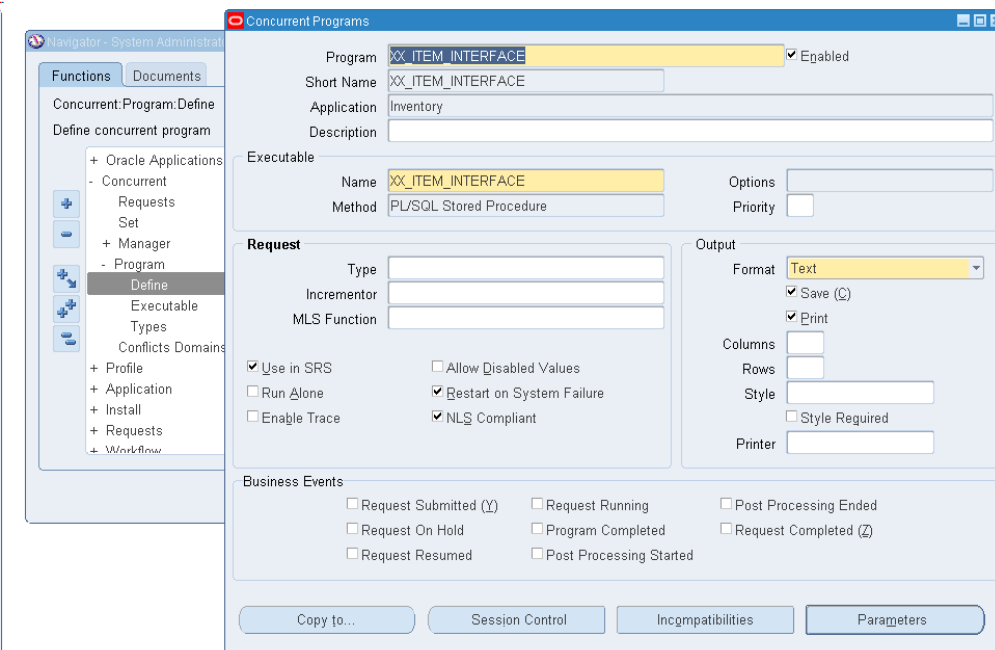
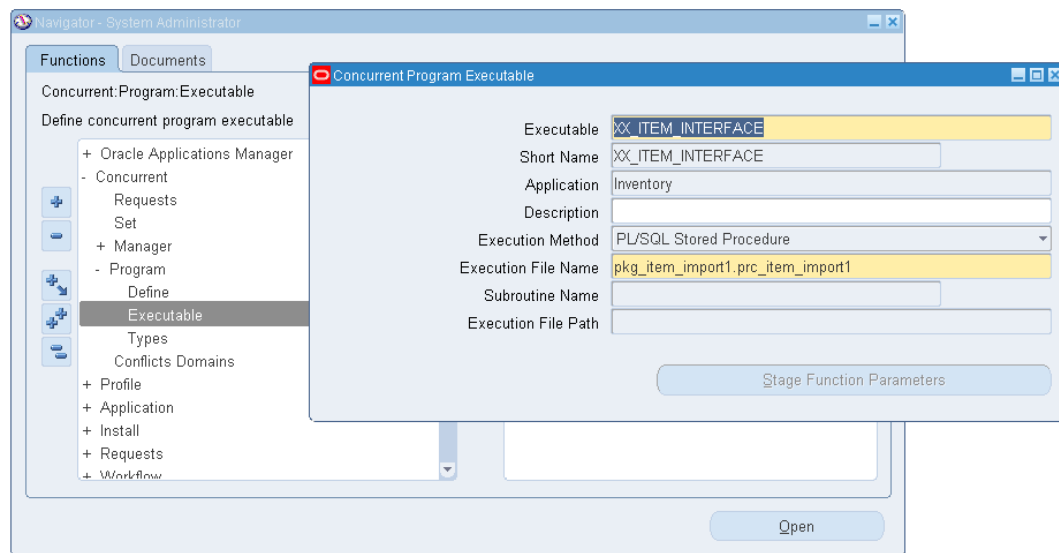
```

1,
1,
1,
1,
NULL,
1
);

COMMIT;
END;
END;
/

```

➤ Create a Executable for Package.



Request Groups

Group: All Inclusive GUI
 Application: Inventory
 Code: INV_ALL_REPORTS_GUI
 Description: All inclusive reports GUI

Requests

Type	Name	Application
Program	Inventory Packing Slip	Inventory
Program	XX_EXC_ITEM	Inventory
Program	XX_ITEM_INTERFACE	Inventory
Program	XX_ITEM_INTERFACE1	Inventory
Program	Overhead Report	Bills of Material
Program	Elemental Inventory Value Report - by Sub	Bills of Material
Program	Inventory Value Report - by Subinventory	Bills of Material
Program	All Inventories Value Report	Bills of Material
Program	Intransit Value Report	Bills of Material
Program	Subinventory Account Value Report	Bills of Material

Description:

Requests

Refresh Data Find Requests Submit a New Request...

Request ID	Name	Parent	Phase	Status	Parameters
5947794	To submit item import from		Completed	Warning	, 1, 1, 1, 1, 1
5947793	XX_ITEM_INTERFACE		Completed	Normal	
5947792	XX_EXC_ITEM		Completed	Normal	D:\ORACLE\VISAPPS\APPS
5947791	XX_EXC_ITEM		Completed	Error	D:\ORACLE\VISAPPS\APPS
5947789	XX_ITEM_INTERFACE		Completed	Normal	
5947784	XX_ITEM_INTERFACE1		Completed	Normal	D:\ORACLE\VISAPPS\APPS
5947783	XX_ITEM_INTERFACE1		Completed	Normal	D:\ORACLE\VISAPPS\APPS
5947782	XX_ITEM_INTERFACE1		Completed	Normal	D:\ORACLE\VISAPPS\APPS
5947780	XX_ITEM_INTERFACE1		Completed	Normal	D:\ORACLE\VISAPPS\APPS
5947779	XX_ITEM_INTERFACE1		Completed	Warning	D:\ORACLE\VISAPPS\APPS

Hgld Request View Details... View Output

Cancel Request Diagnostics View Log...

Or else RUN the Package in Backend :

SET SERVEROUTPUT ON;

declare

a varchar2(32767);

b number;

begin

APPS.pkg_item_import1.prc_item_import1(a,b);

dbms_output.put_line(a || b);

end;

OUTPUT:

anonymous block completed

ENTER THE LOOP

1. V1

THE CODE ORG CODE IS V1

N_SEGMENT1: 1

1. V1

THE CODE ORG CODE IS V1

N_SEGMENT1: 2

1. V1

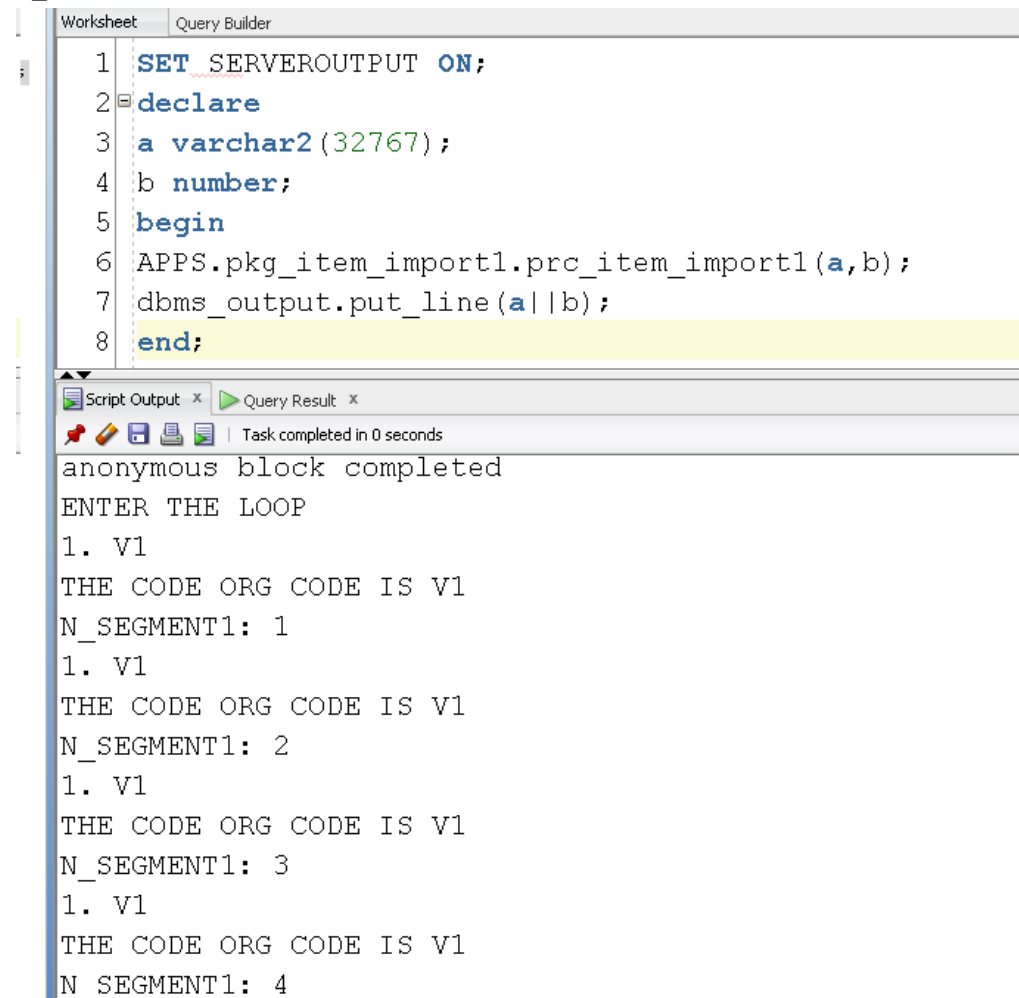
THE CODE ORG CODE IS V1

N_SEGMENT1: 3

1. V1

THE CODE ORG CODE IS V1

N_SEGMENT1: 4



The screenshot displays the Oracle SQL Developer environment. The top pane, titled 'Worksheet', contains a PL/SQL script with the following lines:

```
1 SET SERVEROUTPUT ON;
2 declare
3   a varchar2(32767);
4   b number;
5 begin
6   APPS.pkg_item_import1.prc_item_import1(a,b);
7   dbms_output.put_line(a||b);
8 end;
```

The bottom pane shows the 'Script Output' window, which displays the execution results of the script. The output is as follows:

```
anonymous block completed
ENTER THE LOOP
1. V1
THE CODE ORG CODE IS V1
N_SEGMENT1: 1
1. V1
THE CODE ORG CODE IS V1
N_SEGMENT1: 2
1. V1
THE CODE ORG CODE IS V1
N_SEGMENT1: 3
1. V1
THE CODE ORG CODE IS V1
N_SEGMENT1: 4
```

```
select * from mtl_system_items_interface
where SEGMENT1 in ('909090','909091','909092','909093');
```

```
22 select * from mtl_system_items_interface
23 where SEGMENT1 in ('909090','909091','909092','909093');
```

	TEMPLATE_NAME	ORGANI...	PROCES...	DESCRIPTION	SEGMENT1	INVENT...	ORGANI...	LAST_UPDATE_DATE	LAST_U...	CREATI...	CREATE...
1	Purchased Item V1		1	Tower	909090	(null)	20406-FEB-24	-106-...	-1	(
2	Purchased Item V1		1	Leq	909091	(null)	20406-FEB-24	-106-...	-1	(
3	Purchased Item V1		1	Bracing	909092	(null)	20406-FEB-24	-106-...	-1	(
4	Purchased Item V1		1	Stub	909093	(null)	20406-FEB-24	-106-...	-1	(
5	Purchased Item V1		1	Tower	909090	(null)	20406-FEB-24	-106-...	-1	(
6	Purchased Item V1		1	Leq	909091	(null)	20406-FEB-24	-106-...	-1	(
7	Purchased Item V1		1	Bracing	909092	(null)	20406-FEB-24	-106-...	-1	(
8	Purchased Item V1		1	Stub	909093	(null)	20406-FEB-24	-106-...	-1	(

```
SELECT * from mtl_system_items_b
where SEGMENT1 in ('909090','909091','909092','909093');
```

```
38
39 SELECT * from mtl_system_items_b
40 where SEGMENT1 in ('909090','909091','909092','909093');
```

	SEGMENT1	DESCRIPTION	INVENTORY_ITEM_ID	ORGANIZATION...	LAST_UPDATE_DATE	LAST_UPDATED_BY	CREATI...	CREATE...	LAST_UPDATE_LOGIN	SUMMA...	ENABLE...
1	909090	Tower	251956	20406-FEB-24	131806-...	1318	5632825N	Y	(
2	909091	Leq	251957	20406-FEB-24	131806-...	1318	5632825N	Y	(
3	909092	Bracing	251958	20406-FEB-24	131806-...	1318	5632825N	Y	(
4	909093	Stub	251959	20406-FEB-24	131806-...	1318	5632825N	Y	(

Then in Staging table Process Flag will be changed when the data is imported to interface table.

```
29
30 SELECT * FROM item_staging_tbl;
```

	SEGMENT1	DESCRIPTION	TEMPLATE_NAME	ORGANIZATION_CODE	PROCESS_FLAG	ERROR_MESSAGE	CREATION_DATE	CREATED_BY	LAST_UPDATE_DATE	LAST_UPDATED_BY
1	909090	Tower	Purchased Item V1		S	(null)	06-FEB-24	-106-FEB-24		-1
2	909091	Leq	Purchased Item V1		S	(null)	06-FEB-24	-106-FEB-24		-1
3	909092	Bracing	Purchased Item V1		S	(null)	06-FEB-24	-106-FEB-24		-1
4	909093	Stub	Purchased Item V1		S	(null)	06-FEB-24	-106-FEB-24		-1