Step 1 stagging table craetion

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 by using sql loader load data into staging table

step 3 develop plsql package staging table into interface table with validations

---------------- package -------------------------------

CREATE OR REPLACE PACKAGE pkg\_item\_import

IS

PROCEDURE prc\_item\_import (errbuf OUT VARCHAR2, retcode OUT NUMBER);

END;

/

*------------------------------------------PAckage Body-----------------------------------------------*

create or replace PACKAGE BODY pkg\_item\_import

IS

PROCEDURE prc\_item\_import (errbuf OUT VARCHAR2, retcode OUT NUMBER)

IS

n\_segment1 VARCHAR2 (40);

n\_organization\_id NUMBER;

lv\_template\_name VARCHAR2 (40);

n\_error\_count NUMBER := 0;

g\_error\_message VARCHAR2 (2000) := '';

n\_record\_count NUMBER := 0;

n\_master\_count NUMBER := 0;

n\_child\_count NUMBER := 0;

n\_request\_id NUMBER := 0;

xx\_org VARCHAR2 (40) := NULL;

no\_org NUMBER := 0;

BEGIN

FOR i IN (SELECT a.ROWID row\_id, a.\*

FROM item\_staging\_tbl a

WHERE a.process\_flag = 'N')

LOOP

no\_org := 0;

n\_master\_count := 0;

***DBMS\_OUTPUT.put\_line*** ('1. ' || i.organization\_code);

SELECT organization\_id

INTO n\_organization\_id

FROM mtl\_parameters

WHERE organization\_code = i.organization\_code;

*--- select a.organization\_id from mtl\_parameters a,item\_staging\_tbl b where a.ORGANIZATION\_CODE=b.ORGANIZATION\_CODE*

FOR ix IN (SELECT organization\_code

FROM item\_staging\_tbl xx

WHERE xx.segment1 = i.segment1)

LOOP

IF n\_master\_count <> 0

THEN

CONTINUE;

ELSE

***DBMS\_OUTPUT.put\_line*** ('2. ' || ix.organization\_code);

n\_error\_count := 0;

n\_segment1 := 0;

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);

EXCEPTION

WHEN OTHERS

THEN

n\_error\_count := 1;

END;

***DBMS\_OUTPUT.put\_line*** ('3. ' || xx\_org);

IF xx\_org = ix.organization\_code

THEN

no\_org := 1;

***DBMS\_OUTPUT.put\_line*** ('Entered if');

lv\_template\_name := NULL;

n\_segment1 := NULL;

n\_record\_count := n\_record\_count + 1;

*--------------------------------Item Code is Existed or Not----------------------------------*

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

n\_error\_count := 1;

g\_error\_message := 'Item Code is Existed';

***DBMS\_OUTPUT.put\_line*** ('N\_SEGMENT1: ' || n\_error\_count);

END IF;

EXCEPTION

WHEN OTHERS

THEN

n\_error\_count := 0;

NULL;

END;

*----------------------------------------Tempalate Name is Existed or not-----------------------------------------*

BEGIN

SELECT template\_name

INTO lv\_template\_name

FROM mtl\_item\_templates

WHERE template\_name = i.template\_name;

EXCEPTION

WHEN OTHERS

THEN

n\_error\_count := 1;

g\_error\_message :=

g\_error\_message

|| ','

|| 'Template name is not existed';

***DBMS\_OUTPUT.put\_line*** ('n\_template: ' || n\_error\_count);

END;

*------------------------------------------------------------------------------------------------------*

***DBMS\_OUTPUT.put\_line*** ('ERROR COUNT: ' || n\_error\_count);

IF n\_error\_count <> 0

THEN

CONTINUE;

ELSE

n\_master\_count := n\_master\_count + 1;

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;

END IF;

END IF; *-- organization verification ends here*

END IF; *-- record insertion verification ends here*

END LOOP;

IF n\_master\_count = 0

THEN

UPDATE item\_staging\_tbl

SET process\_flag = 'E',

error\_message = g\_error\_message

WHERE ROWID = i.row\_id;

END IF;

IF no\_org = 0

THEN

UPDATE item\_staging\_tbl

SET process\_flag = 'E',

error\_message =

error\_message || ' ' || 'This item has no Master Org'

WHERE ROWID = i.row\_id;

END IF;

END LOOP;

IF n\_record\_count = 0

THEN

retcode := 1;

END IF;

fnd\_file.put\_line

(fnd\_file.LOG,

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Covad Item Import Interface program started\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

);

fnd\_file.put\_line (fnd\_file.LOG,

'Total master records processed: ' || n\_master\_count

);

fnd\_file.put\_line (fnd\_file.LOG,

'Total Child records processed: ' || n\_child\_count

);

fnd\_file.put\_line

(fnd\_file.LOG,

'\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Covad Item Import Interface program ended\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*'

);

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 prc\_item\_import;

END pkg\_item\_import;

/

----------------------end conversion---------------------------

Step 4

Execute from back end

declare

a varchar2(32767);

b number;

begin

pkg\_item\_import.prc\_item\_import(a,b);

***dbms\_output.put\_line***(a||b);

end;

/

-------------------- ANOTHER METHOD --------------------

CREATE OR REPLACE PACKAGE BODY pkg\_item\_import.prc\_item\_import

IS

g\_user\_id fnd\_user.user\_id%TYPE;

g\_login\_id NUMBER (15) := 0;

PROCEDURE prc\_item\_import (errbuf OUT VARCHAR2, retcode OUT VARCHAR2)

IS

CURSOR c1

IS

SELECT a.ROWID row\_id, a.\*

FROM item\_staging\_tbl a

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

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 INVLID ';

***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;

IF i.template\_name IS NOT NULL

THEN

*----------------------------------------Tempalate Name is Existed or not-----------------------------------------*

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;

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;

/