**Introduction/ Issue: *This interface is developed to cater the requirement of BOM import from flat file.***

**Why we need to do / Cause of the issue: *In Order to outsource information from any third party system, we need to write an appropriate interface program. Here we have developed a BoM interface to interface the BoM quickly.***

**How do we solve: *Use this code to leverage the script and make necessary changes, if any.***

|  |
| --- |
| CREATE TABLE xx\_doyen\_bom\_stg  (  org\_code VARCHAR2(30),  asly\_code VARCHAR2(240),  comp\_code VARCHAR2(240),  qty NUMBER,  subinventory VARCHAR2(100),  uom VARCHAR2(30),  status VARCHAR2(30),  status\_msg VARCHAR2(2000)  ); CREATE OR REPLACE PROCEDURE XX\_DOYEN\_BOM\_STG\_API AS  -- API input variables  l\_bom\_header\_rec bom\_bo\_pub.bom\_head\_rec\_type := bom\_bo\_pub.g\_miss\_bom\_header\_rec;  l\_bom\_revision\_tbl bom\_bo\_pub.bom\_revision\_tbl\_type := bom\_bo\_pub.g\_miss\_bom\_revision\_tbl;  l\_bom\_component\_tbl bom\_bo\_pub.bom\_comps\_tbl\_type := bom\_bo\_pub.g\_miss\_bom\_component\_tbl;  l\_bom\_ref\_designator\_tbl bom\_bo\_pub.bom\_ref\_designator\_tbl\_type := bom\_bo\_pub.g\_miss\_bom\_ref\_designator\_tbl;  l\_bom\_sub\_component\_tbl bom\_bo\_pub.bom\_sub\_component\_tbl\_type := bom\_bo\_pub.g\_miss\_bom\_sub\_component\_tbl;  -- API output variables  x\_bom\_header\_rec bom\_bo\_pub.bom\_head\_rec\_type := bom\_bo\_pub.g\_miss\_bom\_header\_rec;  x\_bom\_revision\_tbl bom\_bo\_pub.bom\_revision\_tbl\_type := bom\_bo\_pub.g\_miss\_bom\_revision\_tbl;  x\_bom\_component\_tbl bom\_bo\_pub.bom\_comps\_tbl\_type := bom\_bo\_pub.g\_miss\_bom\_component\_tbl;  x\_bom\_ref\_designator\_tbl bom\_bo\_pub.bom\_ref\_designator\_tbl\_type := bom\_bo\_pub.g\_miss\_bom\_ref\_designator\_tbl;  x\_bom\_sub\_component\_tbl bom\_bo\_pub.bom\_sub\_component\_tbl\_type := bom\_bo\_pub.g\_miss\_bom\_sub\_component\_tbl;  x\_message\_list error\_handler.error\_tbl\_type;  l\_error\_table error\_handler.error\_tbl\_type;  l\_output\_dir VARCHAR2(500) := '/usr/tmp/test';  l\_debug\_filename VARCHAR2(60) := 'test\_debug.dbg';  l\_return\_status VARCHAR2(1) := NULL;  l\_msg\_count NUMBER := 0;  l\_cnt NUMBER := 1;  -- WHO columns  l\_user\_id NUMBER := -1;  l\_resp\_id NUMBER := -1;  l\_application\_id NUMBER := -1;  l\_row\_cnt NUMBER := 1;  l\_user\_name VARCHAR2(30) := 'DOYENSYS';  l\_resp\_name VARCHAR2(30) := 'BILLS OF MATERIAL';  l\_item\_seq NUMBER; BEGIN  -- Get the user\_id  SELECT user\_id  INTO l\_user\_id  FROM fnd\_user  WHERE user\_name = l\_user\_name;  -- Get the application\_id and responsibility\_id  SELECT application\_id,  responsibility\_id  INTO l\_application\_id, l\_resp\_id  FROM fnd\_responsibility  WHERE responsibility\_key = l\_resp\_name;  fnd\_global.Apps\_initialize(l\_user\_id, l\_resp\_id, l\_application\_id);  dbms\_output.Put\_line('Initialized applications context: '  || l\_user\_id  || ' '  || l\_resp\_id  ||' '  || l\_application\_id);  FOR ch IN (SELECT DISTINCT asly\_code,  comp\_code,  org\_code  FROM xx\_doyen\_bom\_stg A,  mtl\_parameters B,  mtl\_system\_items C,  mtl\_system\_items d  WHERE A.org\_code = B.organization\_code  AND A.asly\_code = C.segment1  AND B.organization\_id = C.organization\_id  AND a.comp\_code = d.segment1  AND d.organization\_id = C.organization\_id  AND C.bom\_enabled\_flag = 'Y'  AND NOT EXISTS (SELECT 1  FROM bom\_bill\_of\_materials X  WHERE  X.assembly\_item\_id = C.inventory\_item\_id  AND X.organization\_id = C.organization\_id)  ) LOOP  l\_bom\_revision\_tbl.DELETE;  l\_bom\_component\_tbl.DELETE;  l\_bom\_ref\_designator\_tbl.DELETE;  l\_bom\_sub\_component\_tbl.DELETE;  x\_bom\_revision\_tbl.DELETE;  x\_bom\_component\_tbl.DELETE;  x\_bom\_ref\_designator\_tbl.DELETE;  x\_bom\_sub\_component\_tbl.DELETE;  -- initialize BOM header  l\_bom\_header\_rec := bom\_bo\_pub.g\_miss\_bom\_header\_rec;  l\_bom\_revision\_tbl := bom\_bo\_pub.g\_miss\_bom\_revision\_tbl;  l\_bom\_component\_tbl := bom\_bo\_pub.g\_miss\_bom\_component\_tbl;  l\_bom\_ref\_designator\_tbl := bom\_bo\_pub.g\_miss\_bom\_ref\_designator\_tbl;  l\_bom\_sub\_component\_tbl := bom\_bo\_pub.g\_miss\_bom\_sub\_component\_tbl;  -- API output variables  x\_bom\_header\_rec := bom\_bo\_pub.g\_miss\_bom\_header\_rec;  x\_bom\_revision\_tbl := bom\_bo\_pub.g\_miss\_bom\_revision\_tbl;  x\_bom\_component\_tbl := bom\_bo\_pub.g\_miss\_bom\_component\_tbl;  x\_bom\_ref\_designator\_tbl := bom\_bo\_pub.g\_miss\_bom\_ref\_designator\_tbl;  x\_bom\_sub\_component\_tbl := bom\_bo\_pub.g\_miss\_bom\_sub\_component\_tbl;  l\_bom\_header\_rec.assembly\_item\_name := ch.asly\_code;  l\_bom\_header\_rec.organization\_code := ch.org\_code;  l\_bom\_header\_rec.assembly\_type := 1;  l\_bom\_header\_rec.transaction\_type := 'CREATE';  l\_bom\_header\_rec.return\_status := NULL;  l\_cnt := 0;  l\_item\_seq := 0;  FOR cl IN (SELECT A.comp\_code,  A.uom,  A.org\_code,  A.subinventory,  A.qty  FROM xx\_doyen\_bom\_stg A,  mtl\_parameters B,  mtl\_system\_items C  WHERE A.org\_code = B.organization\_code  AND A.comp\_code = C.segment1  AND B.organization\_id = C.organization\_id  AND A.asly\_code = ch.asly\_code  AND A.org\_code = ch.org\_code) LOOP  l\_cnt := l\_cnt + 1;  l\_item\_seq := l\_item\_seq + 10;  L\_bom\_component\_tbl (l\_cnt).organization\_code := ch.org\_code;  L\_bom\_component\_tbl (l\_cnt).assembly\_item\_name := ch.asly\_code;  L\_bom\_component\_tbl (l\_cnt).start\_effective\_date := SYSDATE;  L\_bom\_component\_tbl (l\_cnt).component\_item\_name := cl.comp\_code;  L\_bom\_component\_tbl (l\_cnt).alternate\_bom\_code := NULL;  L\_bom\_component\_tbl (l\_cnt).supply\_subinventory := cl.subinventory;  L\_bom\_component\_tbl (l\_cnt).location\_name := NULL;  L\_bom\_component\_tbl (l\_cnt).comments := 'Created from BOM API';  L\_bom\_component\_tbl (l\_cnt).item\_sequence\_number := l\_item\_seq;  L\_bom\_component\_tbl (l\_cnt).operation\_sequence\_number := l\_item\_seq;  L\_bom\_component\_tbl (l\_cnt).transaction\_type := 'CREATE';  L\_bom\_component\_tbl (l\_cnt).quantity\_per\_assembly := cl.qty;  L\_bom\_component\_tbl (l\_cnt).return\_status := NULL;  END LOOP;  error\_handler.initialize;  DBMS\_OUTPUT.PUT\_LINE('=======================================================');  DBMS\_OUTPUT.PUT\_LINE('Calling Bom\_Bo\_Pub.process\_bom API');  bom\_bo\_pub.Process\_bom (p\_bo\_identifier => 'BOM',  p\_api\_version\_number => 1.0,  p\_init\_msg\_list => TRUE, p\_bom\_header\_rec => l\_bom\_header\_rec,  p\_bom\_revision\_tbl => l\_bom\_revision\_tbl,  p\_bom\_component\_tbl => l\_bom\_component\_tbl,  p\_bom\_ref\_designator\_tbl => l\_bom\_ref\_designator\_tbl,  p\_bom\_sub\_component\_tbl => l\_bom\_sub\_component\_tbl,  x\_bom\_header\_rec => x\_bom\_header\_rec,  x\_bom\_revision\_tbl => x\_bom\_revision\_tbl,  x\_bom\_component\_tbl => x\_bom\_component\_tbl,  x\_bom\_ref\_designator\_tbl => x\_bom\_ref\_designator\_tbl,  x\_bom\_sub\_component\_tbl => x\_bom\_sub\_component\_tbl,  x\_return\_status => l\_return\_status, x\_msg\_count => l\_msg\_count,  p\_debug => 'N'  );  DBMS\_OUTPUT.PUT\_LINE('=======================================================');  dbms\_output.Put\_line('Return Status: ' ||l\_return\_status);  COMMIT;  END LOOP; EXCEPTION  WHEN OTHERS THEN  dbms\_output.Put\_line('Exception Occured :');  dbms\_output.Put\_line(SQLCODE ||':' ||SQLERRM);  DBMS\_OUTPUT.PUT\_LINE('======================================================='); RAISE; END XX\_DOYEN\_BOM\_STG\_API; |

**Conclusion:**

**This code will help you to get import the BoM.**