

|  |
| --- |
|  |
| Approach Document  CBS DataStage Migration from version 8.7 to version 11.x |
|  |
|  |

Strictly Private, Proprietary and Confidential  
For Distribution to Authorized Personnel Only

Table of Contents

[1. Purpose 2](#_Toc470066578)

[2. Scope 2](#_Toc470066579)

[3. Identify The List Of Deprecated Stages in 11.x 2](#_Toc470066580)

[4. Identify The List Of Deprecated Stages In CBS Application 3](#_Toc470066581)

[5. Process Of Replacing Oracle Enterprise Stages With Connector Stage 4](#_Toc470066582)

[6. Build 10](#_Toc470066583)

[7. Testing Methodology 10](#_Toc470066584)

[8. Document Revision History 14](#_Toc470066585)

1. Purpose

CBS Application uses DataStage 8.7 as the ETL tool for processing the claims data. As there is a planned upgrade from DataStage 8.7 to Datastage11.x, this document outlines the systematic approach that can be followed for migration of the existing components to the new environment.

1. Scope

The scope of this document is as below—

* Determine the list of components/stages that are deprecated in DS 11.x
* From the above list, identify the set of components in CBS (both ODS and BNP projects) that are impacted
* Explain the process of identifying the jobs that need to be connector migrated
* Elaborate the automated connector migration process through IBM’s connector migration tool
* Define the testing strategy
* Solutions that can help in reducing the migration effort

1. Identify The List Of Deprecated Stages in 11.x

Below table displays the list of stages deprecated in 11.x and their replacements as recommended by IBM.

|  |  |
| --- | --- |
| **Deprecated stage Name** | **Recommended stage Name** |
| DB2Z | DB2 Connector |
| DB2 UDB API |
| DB2 UDB Enterprise |
| DB2 UDB Load |
| DRS | DRS Connector |
| Dynamic RDBMS | DB2 Connector |
| Oracle Connector |
| ODBC Connector |
| Java Client | Java Integration stage |
| Java Transformer |
| Netezza Enterprise | Netezza Connector |
| ODBC Enterprise | ODBC Connector |
| Oracle 7 Load | Oracle Connector |
| Oracle OCI Load |
| Oracle Enterprise |
| Teradata API | Teradata Connector |
| Teradata Enterprise |
| Teradata Load |
| Teradata Multiload |
| WebSphere MQ | WebSphere MQ Connector |

Source of Information:- <http://www.ibm.com/support/knowledgecenter/SSZJPZ_11.3.0/com.ibm.swg.im.iis.conn.migtool.doc/topics/removal_stages_palette.html>

1. Identify The List Of Deprecated Stages In CBS Application

CBS application has two DataStage projects (in DEV)

1) EGVDVCRXSTG01/whs\_pbmdss\_ods\_dev

2) EGVDVCRXSTG01/whs\_pbm\_billing\_payment\_dev\_rec1

Both of these projects have jobs which are using Oracle Enterprise stage either as source table or reference table or as target table.

As per the recommendation by IBM, while migrating the jobs from DataStage 8.7 version to DataStage11.x version, Oracle Enterprise stage needs to be replaced with Oracle connector Stage.

* List of jobs for EGVDVCRXSTG01/whs\_pbmdss\_ods\_dev project that need to be connector migrated have been identified and is attached below.



For each of the jobs, their respective master sequences also have been identified and mentioned in the above excel which will help during the testing phase.

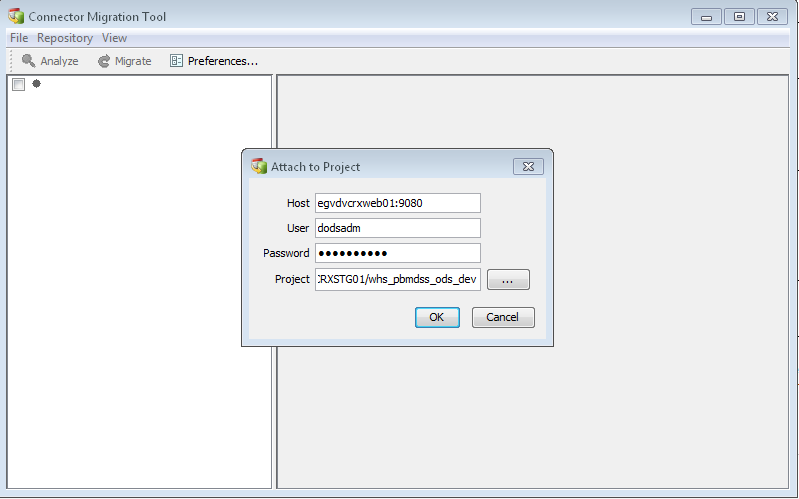
* Inventory list of jobs for EGVDVCRXSTG01/ whs\_pbm\_billing\_payment\_dev\_rec1 that need to be connector migrated



1. Process Of Replacing Oracle Enterprise Stages With Connector Stage

For identifying the jobs which use Oracle Enterprise stage, IBM recommends to use [Connector Migration Tool](http://www.ibm.com/support/knowledgecenter/en/SSZJPZ_11.3.0/com.ibm.swg.im.iis.conn.migtool.doc/topics/connectormigrationtool_intro.html?view=kc). The process to migrate the Oracle Enterprise stage to Oracle Connector stage using connector migration tool is explained below.

* Login to IBM Infosphere Connector Migration Tool and provide the necessary project credential.



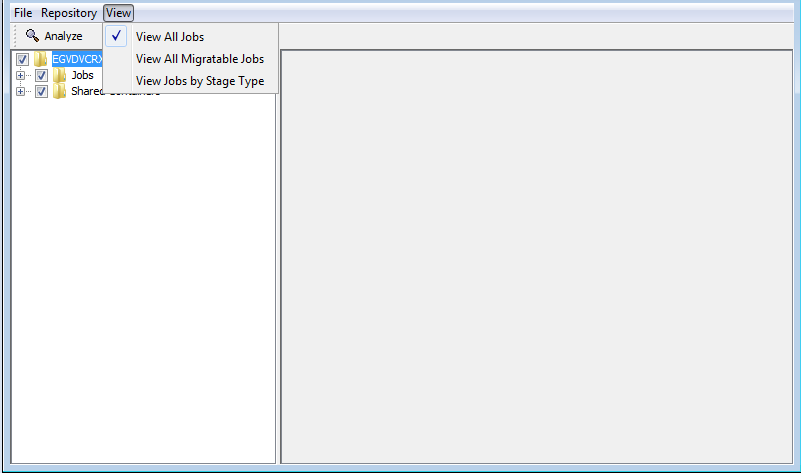
* Click on View and you will see 3 options:

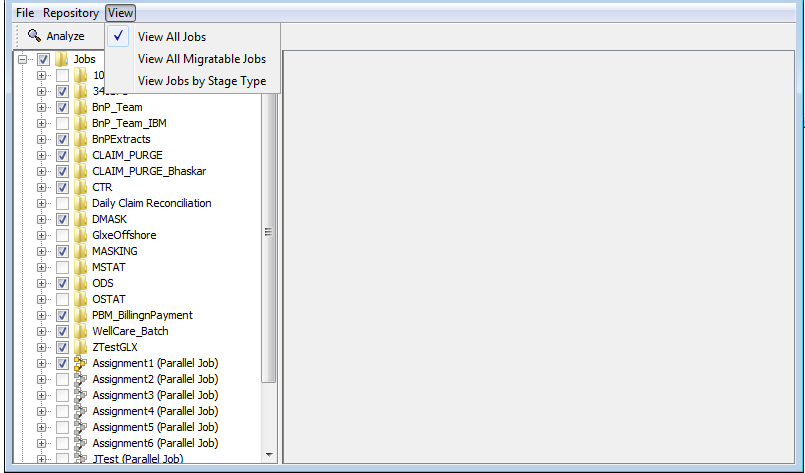
1) View all Jobs

2) View all migratable jobs

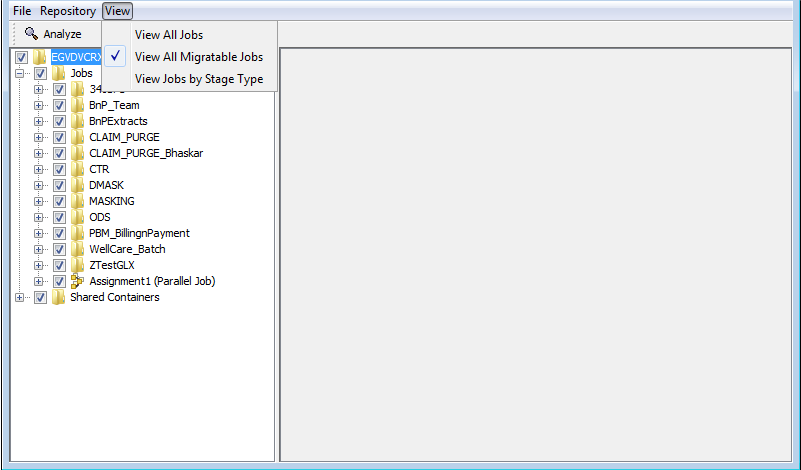
3) View jobs by stage types

* View all Jobs: - View all jobs will display all the jobs of Datastage project irrespective of jobs are migratable or not.

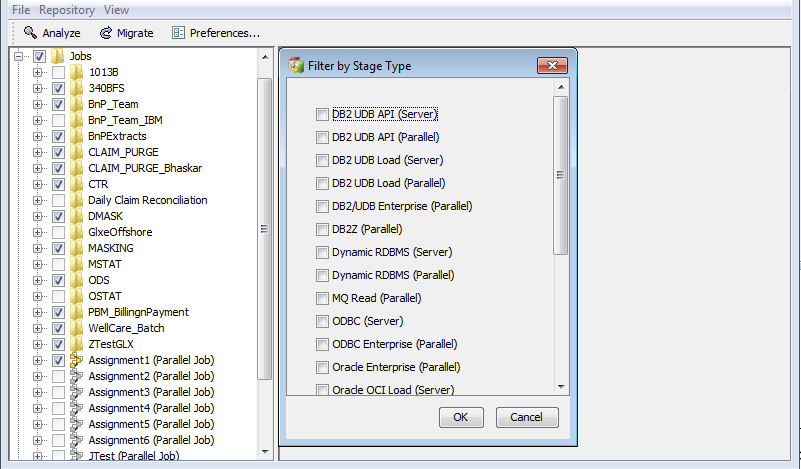




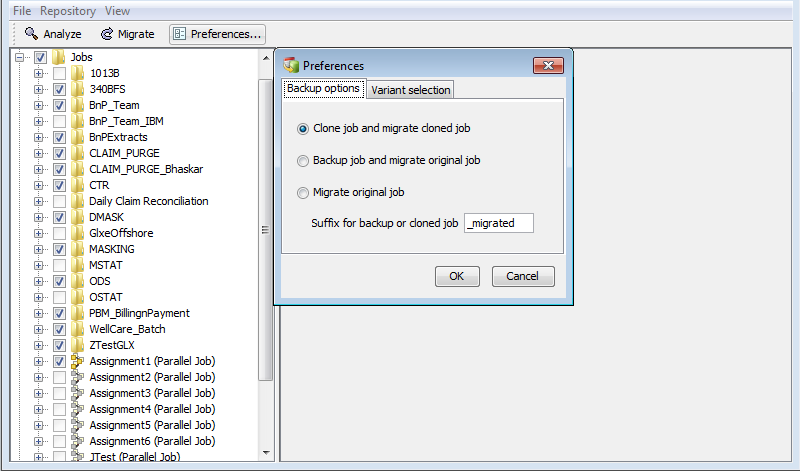
* View all migratable jobs: - Will display list of jobs which are using Oracle connector stage and ready to be migrated.



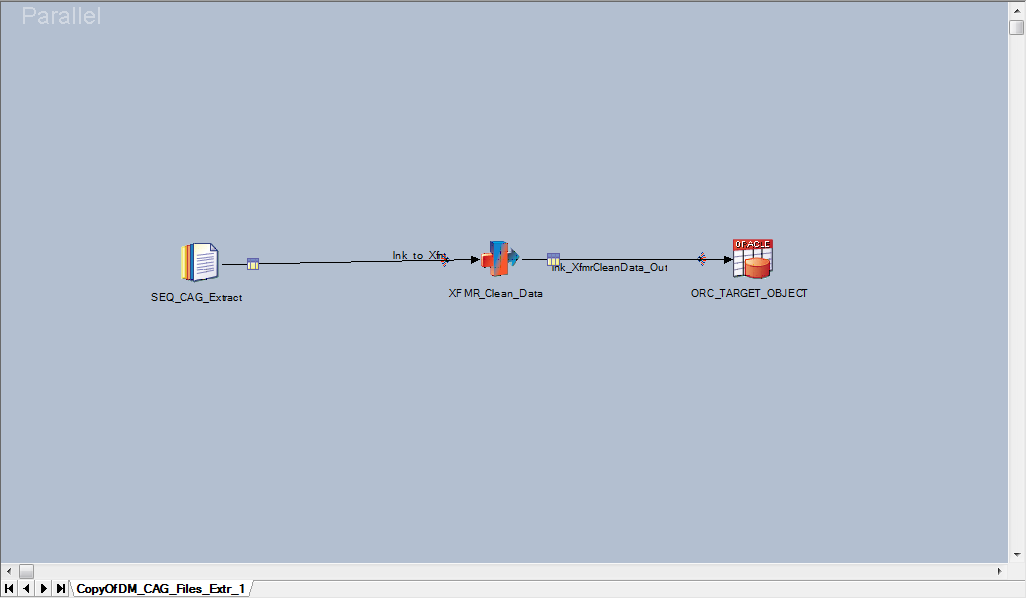
* View jobs by stage types: - Will display the list of jobs using deprecated stage and allow choosing the deprecated stage (ex: Oracle Enterprise stage for CBS).



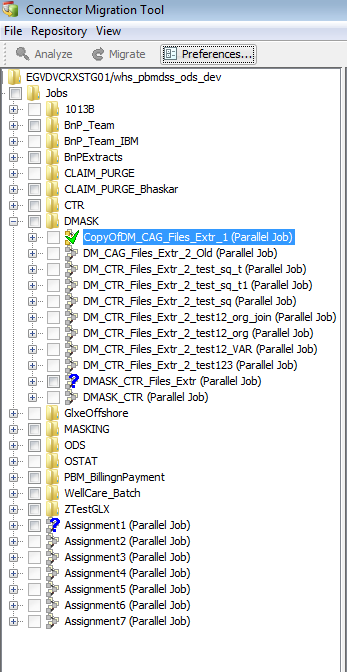
* Click Preferences and choose how to migrate the job:-
* Choose Clone and migrate cloned job to make a copy of the job and then migrate the copy. The original job remains intact.
* Choose back up job and migrate original job to make a copy of the job and then migrate the original job.
* Choose Migrate original job to migrate the job without making a backup.



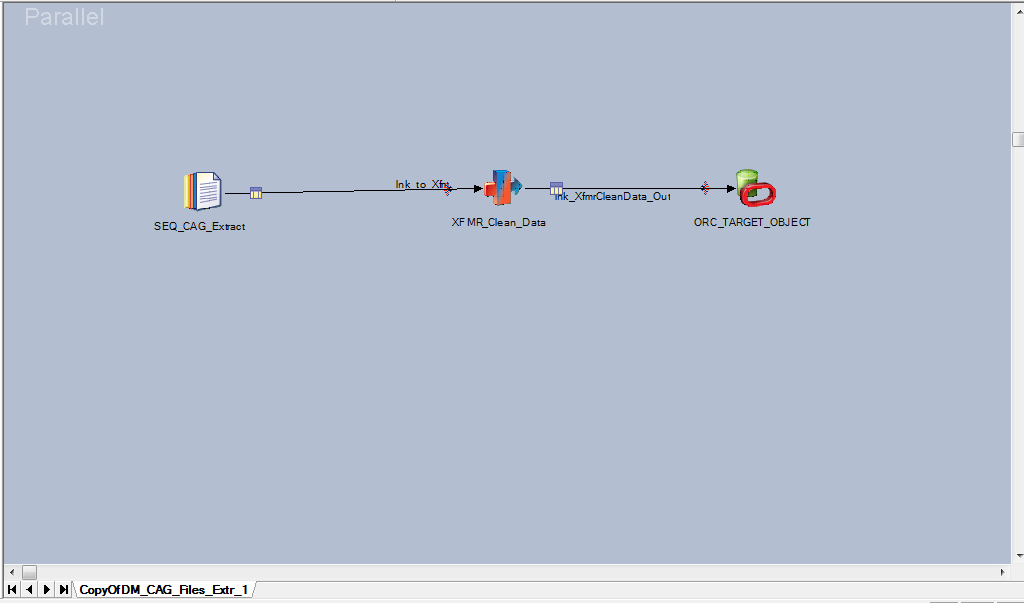
* Job before Migration:-



* Once you login to connector migration tool you will find the below screen, on opening the folder you will find blue question mark for jobs to be migrated. Then when you click the migrate button the connector tool automatically converts the enterprise oracle stages to oracle connector stage and the tick mark turns green.



* Job after Migration: -



1. Build

* Take a backup of the DataStage project for which migration is to be done.
* Create a new DataStage project as project B with the help of admin.
* Import the DataStage project backup to project B.
* Use Connector migration tool to migrate the Oracle enterprise stages to Oracle connector stages in project B.
* Check the jobs if the enterprise stages are converted to Oracle connector stage and compile the respective jobs and sequences.
* Create a new schema with help of DBA and create the replica of target table structure with necessary privileges which will be impacted.
* Update the schema name parameter in parameter set for project B to refer to the new schema.

1. Testing Methodology

* Run the job in project A (with enterprise stage) which loads table in schema A (existing/old).
* Run the same job in project B (with connector stage) which will load table in schema B.
* Once the tables are loaded in both the schemas, run minus queries between them in both the ways (“A-B” & “B-A”).
* Make sure there are no fallouts between the tables in two schemas.
* Verify the count of records loaded into respective tables in both the schemas.

**Example**

* Assume a table by name PHONE\_TEST1 is present in SCHEMA\_A, and its replica PHONE\_TEST2 is present in SCHEMA\_B. Execute the jobs with oracle enterprise stage pointing to SCHEMA A and with Oracle connector stage pointing to SCHEMA B.
* After the jobs get completed, run minus query as below –

**SELECT count(\*)**

**FROM (**

**(**

**SELECT ODS\_PHONE\_SK**

**,SRC\_SYS\_ID**

**,SRC\_PHONE\_ID**

**,SRC\_HLDR\_TYP\_CD**

**,SRC\_HLDR\_ID**

**,PHONE\_TYP\_CD**

**,REC\_EFF\_DTTM**

**,SRC\_SEQ\_NBR**

**,AREA\_CD**

**,PHONE\_NBR**

**,EXTN**

**,REC\_TRMN\_DTTM**

**,REC\_CREATE\_DTTM**

**,REC\_INACTV\_FLAG**

**,LAST\_UPDT\_DTTM**

**,PHONE\_LNK\_TYP\_CD**

**,PHONE\_LNK\_ID**

**,ODS\_PRCS\_BTCH\_DTL\_ID**

**,ODS\_INS\_UPDT\_UID**

**,ODS\_UPD\_DTTM**

**,ODS\_REC\_EFF\_DTTM**

**,ODS\_REC\_END\_DTTM**

**,ODS\_REC\_STAT\_CD**

**,ODS\_INS\_DTTM**

**FROM PHONE\_TEST1**

**Minus**

**SELECT ODS\_PHONE\_SK**

**,SRC\_SYS\_ID**

**,SRC\_PHONE\_ID**

**,SRC\_HLDR\_TYP\_CD**

**,SRC\_HLDR\_ID**

**,PHONE\_TYP\_CD**

**,REC\_EFF\_DTTM**

**,SRC\_SEQ\_NBR**

**,AREA\_CD**

**,PHONE\_NBR**

**,EXTN**

**,REC\_TRMN\_DTTM**

**,REC\_CREATE\_DTTM**

**,REC\_INACTV\_FLAG**

**,LAST\_UPDT\_DTTM**

**,PHONE\_LNK\_TYP\_CD**

**,PHONE\_LNK\_ID**

**,ODS\_PRCS\_BTCH\_DTL\_ID**

**,ODS\_INS\_UPDT\_UID**

**,ODS\_UPD\_DTTM**

**,ODS\_REC\_EFF\_DTTM**

**,ODS\_REC\_END\_DTTM**

**,ODS\_REC\_STAT\_CD**

**,ODS\_INS\_DTTM**

**FROM PHONE\_TEST2**

**)**

**UNION ALL**

**(**

**SELECT ODS\_PHONE\_SK**

**,SRC\_SYS\_ID**

**,SRC\_PHONE\_ID**

**,SRC\_HLDR\_TYP\_CD**

**,SRC\_HLDR\_ID**

**,PHONE\_TYP\_CD**

**,REC\_EFF\_DTTM**

**,SRC\_SEQ\_NBR**

**,AREA\_CD**

**,PHONE\_NBR**

**,EXTN**

**,REC\_TRMN\_DTTM**

**,REC\_CREATE\_DTTM**

**,REC\_INACTV\_FLAG**

**,LAST\_UPDT\_DTTM**

**,PHONE\_LNK\_TYP\_CD**

**,PHONE\_LNK\_ID**

**,ODS\_PRCS\_BTCH\_DTL\_ID**

**,ODS\_INS\_UPDT\_UID**

**,ODS\_UPD\_DTTM**

**,ODS\_REC\_EFF\_DTTM**

**,ODS\_REC\_END\_DTTM**

**,ODS\_REC\_STAT\_CD**

**,ODS\_INS\_DTTM**

**FROM PHONE\_TEST2**

**Minus**

**SELECT ODS\_PHONE\_SK**

**,SRC\_SYS\_ID**

**,SRC\_PHONE\_ID**

**,SRC\_HLDR\_TYP\_CD**

**,SRC\_HLDR\_ID**

**,PHONE\_TYP\_CD**

**,REC\_EFF\_DTTM**

**,SRC\_SEQ\_NBR**

**,AREA\_CD**

**,PHONE\_NBR**

**,EXTN**

**,REC\_TRMN\_DTTM**

**,REC\_CREATE\_DTTM**

**,REC\_INACTV\_FLAG**

**,LAST\_UPDT\_DTTM**

**,PHONE\_LNK\_TYP\_CD**

**,PHONE\_LNK\_ID**

**,ODS\_PRCS\_BTCH\_DTL\_ID**

**,ODS\_INS\_UPDT\_UID**

**,ODS\_UPD\_DTTM**

**,ODS\_REC\_EFF\_DTTM**

**,ODS\_REC\_END\_DTTM**

**,ODS\_REC\_STAT\_CD**

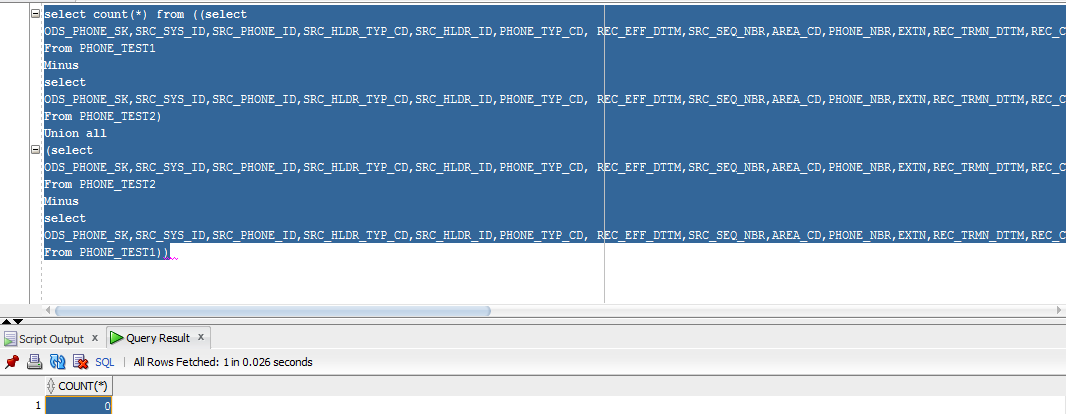
**,ODS\_INS\_DTTM**

**FROM PHONE\_TEST1**

**)**

**)**

The result of the above query should be zero as shown below.



* For Jobs which have dataset or sequential file stage as target, validate the link row count for both the DataStage jobs in project A and project B and it should be same after the jobs completes it run.

**PL/SQL Procedure for Table Comparison:**

Attached below is the generic PL/SQL procedure which will dynamically create and execute minus queries based on the table names passed and will return the results.



Here is what the procedure does—

* + Accepts any 2 table names as parameters
  + Gets the list of columns for these tables from ALL\_TAB\_COLUMNS and excludes the common audit columns.
  + Creates minus queries for these 2 tables dynamically for both ways (A minus B & B minus A)
  + Executes these minus queries and if there are no fallouts/differences, a record is inserted into AUDIT table with result as PASS.
  + If there are any fallouts, the count of fallouts will be written into AUDIT table and result will be FAIL

DDL for the audit table is attached below.



1. Document Revision History

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Version | Created By | Reviewed By | Approved By | Description of change |
| 18-Nov-16 | 0.1 | GalaxE Solutions |  |  | Initial Version |
| 21-Dec-16 | 0.2 | GalaxE Solutions |  |  | Updated with table comparison PLSQL and BNP jobs list for migration |