Conversion

Data Extraction Work stream

ETL Detailed Design Document – Environment Sync UP

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version No.** | **Date** | **Author** | **Revision Description** |
| 0.01 | 06/03/2020 | Zahir Sheikh | Initial Version |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

Table of Contents

[Document History 1](#_Toc42171405)

[**1** **Introduction** 4](#_Toc42171406)

[1.1 Document Outline 4](#_Toc42171407)

[1.2 Purpose of this document 4](#_Toc42171408)

[1.3 Scope 4](#_Toc42171409)

[1.4 Out Of Scope 4](#_Toc42171410)

[1.5 Definitions, Acronyms and Abbreviations 5](#_Toc42171411)

[**2** **System Overview** 5](#_Toc42171412)

[2.1 Systems Involved 5](#_Toc42171413)

[2.2 Assumptions 5](#_Toc42171414)

[2.3 Dependencies/Risks 5](#_Toc42171415)

[2.4 Proposed Solution/HLD 6](#_Toc42171416)

[2.5 Solution Description/LLD 7](#_Toc42171417)

[2.5.1 Pentaho Files Transfer 7](#_Toc42171418)

[2.5.2 Temp\_files Transfer 7](#_Toc42171427)

[2.5.3 Database Objects transfer 8](#_Toc42171428)

[2.5.4 Purging Backup Directory 8](#_Toc42171429)

[2.5.5 Shell Script 9](#_Toc42171430)

[2.5.6 Python Script 9](#_Toc42171431)

[**3** **Framework Execution** 10](#_Toc42171432)

[3.1 Master Job Execution 10](#_Toc42171434)

[3.2 Database Job Execution 11](#_Toc42171435)

[**4** **Database Error** 11](#_Toc42171436)

[4.1 Exception Details 11](#_Toc42171438)

[4.2 Exception Objects DDL 11](#_Toc42171439)

[**5** **Related Documents** 12](#_Toc42171440)

1. **Introduction**

1.1 Document Outline

ABC Project deliver UF files based on bimonthly Single View bill cycles and for every deliverable ABC Team has to sync up the environment based on the modified objects i.e. Scripts; Pentaho – trans and jobs; and database objects – Tables, Indexes, Store Procedures and Functions from the previous delivery environment to proposed environment for individual product type like BI, ME, AV – PRI and SIP, BV and BE.

1.2 Purpose of this document

This design document speaks about the technical implementation of ETL / Data Integration for environment sync up of stage server i.e. APP as well as Database. This Document intended for use by the ABC Migration Project Team. This provides a template, which assists the reader in understanding the requirements, design and development of the Environment Sync Up solution.

1.3 Scope

The scope of this document is:

* Explain the ETL design & architecture for Environment Set Up to avoid human errors and deliver flawless UF files for each bill cycle
  + ETL Design from Source to Target Environment
  + ETL Design of restart ability
  + ETL Design of Audit
  + ETL Design of Error
* Audit for each target database objects caught exceptions while executing the DDL used to ensure no objects has been dropped off.

1.4 Out Of Scope

Following are the out-of-scope items for this requirement/document.

* + Scripts folder for Migration Environment.
  + Clean Up Environment Sync Up but can be used python script “environmentSetup.py” to sync DB objects by passing schema names.

1.5 Definitions, Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| **Terminology** | **Acronym** | **Abbreviation** |
| UNI | UNI | User Network Interface |
| EVC | EVC | Ethernet Virtual Circuit |
| EVPL | EVPL | Ethernet Virtual Private Line |
| EPL | EPL | Ethernet Private Line |
| ENS | ENS | Ethernet Network Service |
| EDI | EDI | Ethernet Dedicated Internet |
| Site Access | - | Network connecting subscriber to service provider |
| EVC Endpoint | - | One end of a virtual circuit |
| Equipment | CPE | Customer Premise Equipment |
| GTI | GTI | Generic Traffic Interface |
| SPS | SPS | Simple Product Specification |

1. **System Overview**

## 2.1 Systems Involved

|  |  |
| --- | --- |
| Stage App Server | To sync up Pentaho – trans & jobs folder as well as temp\_files folder. |
| Database Server | To sync DB objects -- Tables, Indexes, Store Procedure and Functions |

## 2.2 Assumptions

* All necessary infrastructure and access, software and licenses required for Environment Sync Up framework will be provided by xyz.

## 2.3 Dependencies/Risks

* Necessary Software and Licenses – python 3.6, cx\_Oracle7, Oracle Client 11g.
* System Access and Application Access – Stage Application Linux Box.

## 2.4 Proposed Solution/HLD

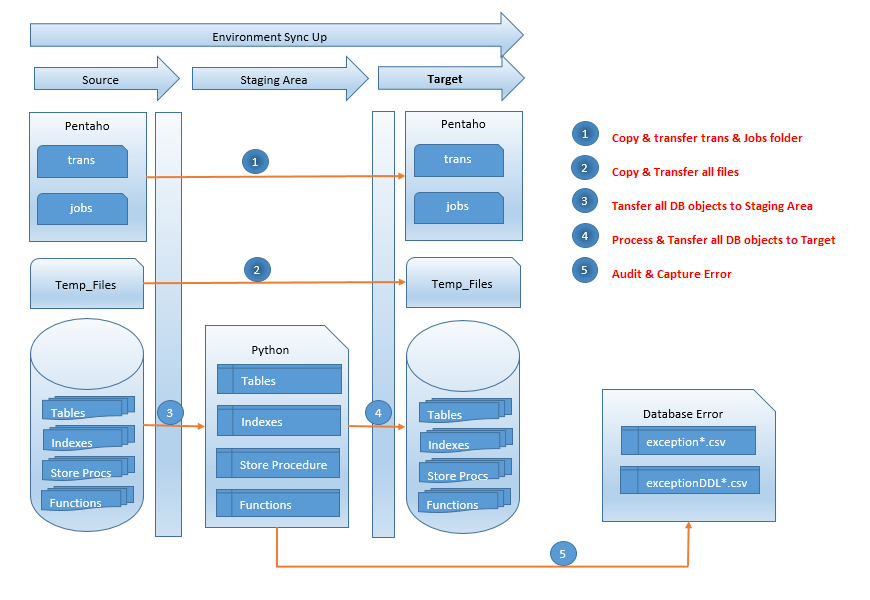


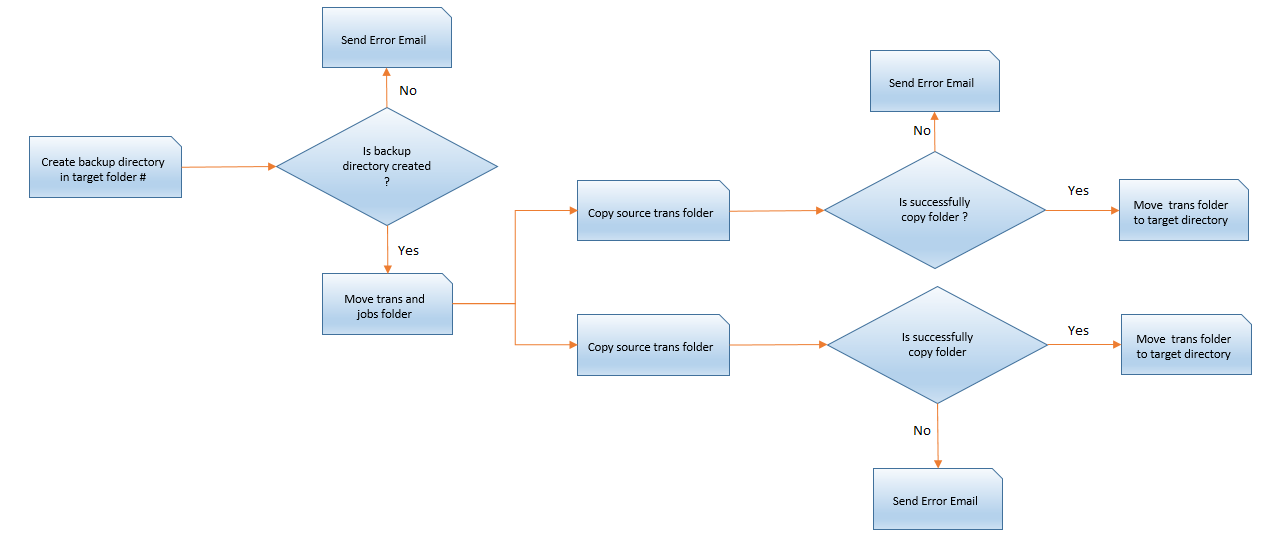
Figure 1 - Proposed Environment Sync up Solution

* Pentaho objects transfer to target directory. These are the sources for Trans and Jobs folders.
* Temp\_Files stuffs transfer to target directory. These are the sources for TT Files, Lookup tables files etc.
* Database objects metadata gets pull from source schema and transfer to target schema.
* Database error put in exception file along with its metadata.

## Solution Description/LLD

### Pentaho Files Transfer

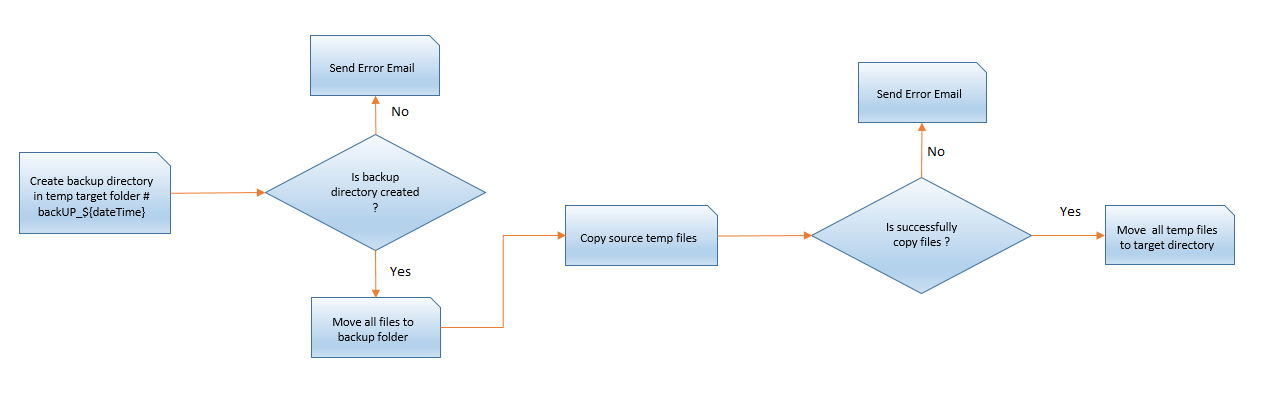
The flow of data for transferring pentaho’s -- trans and jobs directories is shown in the below flowchart.





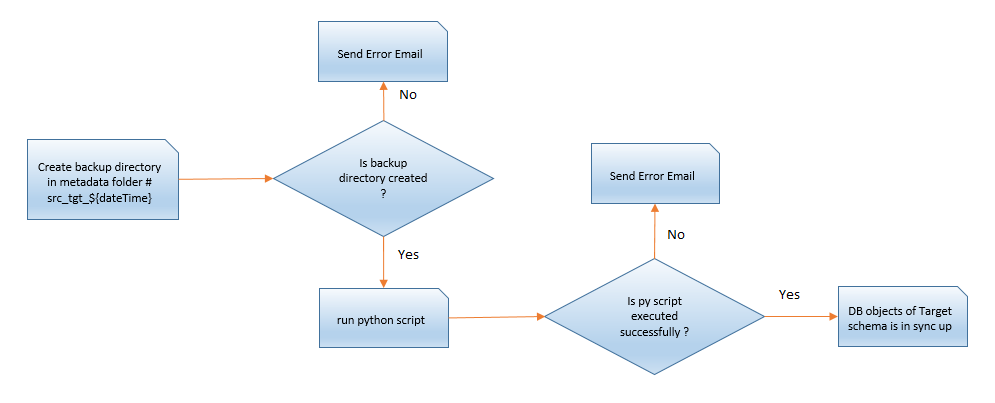
### Temp\_files Transfer

The flow of data for transferring temp files – TT files, Reference files etc. is shown in the below flowchart.



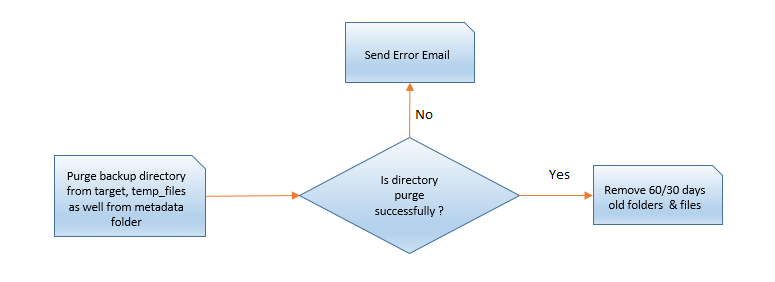
### Database Objects transfer

The flow of data for transferring DB objects – Tables, Indexes, Store Procedure, Functions and Grants is shown in the below flowchart.



### Purging Backup Directory

The flow of purge for removing backup folders – Target, Temp\_files and metadata folder based on 60/30 days old criteria is shown in the below flowchart.



### Shell Script

This framework accept two parameters (i.e. Source Schema and Target Schema ) in order to sync up the database schema by calling the python script, which internally connecting Stage DB with Hardcoded connection parameters.

This script is divided into eight logical section i.e.

1. Exporting Oracle variables to setup the bash environment
2. Assignment to local variables as well as Log file by name ${logDir}/envSetup\_from\*.log
3. Failure function terminate execution of this shell script and sends an email to Support Team with attach log.
4. Create Backup folder in TARGET directory and transfer jobs & trans directories into it. Copy above folders from SOURCE to TARGET folder.
5. Create Backup folder in temp TARGET directory and transfer all files into it. After this copy all the temp files from source to target directory.
6. Create Metadata directory for current sync up process and execute python framework.
7. Purging all the 60/30 days old backup directory and files from TARGET, TEMP\_FILES & METADATA directories respectively.
8. Send the Success email to Support Team once Job is finished

### Python Script

This framework accept three parameters (i.e. Source Schema, Target Schema and metadata directory path --> sourceSchema\_TargetSchema\_datetimestamp) in order to sync up database schema by connecting Stage DB with Hardcoded connection param. This script is divided into five logical section i.e.

1. Setting the Oracle Session Parameters --> This requires to avoid additional information getting pull up in DDL i.e. Storage Type, locations etc.

2. Table Sync up section --> This section is divided furthermore into two subsections

* + Column Mismatch --> It compares the same table in two schemas and find out the column anomaly for example table zahir has 12 columns in source schema where as in target schema it has less/more (i.e. 11/14) columns then it would drop this table from the target schema so that it would be picked up in missing table section.
  + Pushing Missing Tables --> This code consider tables are missing in target schema by comparing with source schema and generates the DDL for such tables. Once metadata DDL gets generated it writes back to csv file in metadata directory by name 'tableMissTgtDF.csv' so that it can be used for audit purpose or reference.

3. Index Sync up section --> This section is divided moreover into two subsections

* + Column Mismatch --> It compares indexes in two schemas and find out the column anomaly for example index zahirIDX has 12 columns in source schema where as in target schema it has less/more (i.e. 11/14) columns then it would drop this index from the target schema so that it would be picked up in missing index section.
  + Pushing Missing Indexes --> This code consider indexes are missing in target schema by comparing with source schema and generates the DDL for such indexes. Once metadata DDL gets generated it writes back to csv file in metadata directory by name 'indexMissTgtDF.csv' so that it can be used for audit purpose or reference.

4. Procedure/Function Sync up section --> This section is divided additionally into two subsections

* + Text Mismatch within line of SP --> It simply count the number of lines of PL/SQL block in two schemas and drop it from the target schema so that it would be picked up in missing Procedure/Function section as logic needs to revisit to find the textual difference within a line. Based on new logic code needs to be rewritten.
  + Pushing Missing PL/SQL Blocks --> This code consider all the PL/SQL blocks are missing in target schema as well as count mismatch for the number of lines by comparing with source schema and generates the PL/SQL Script for such SP. Once metadata PL/SQL block gets generated, it writes back to csv file in metadata directory by name 'storeprocMissTgtDF.csv' so that it can be used for audit purpose or reference.

5. Grant section --> This section is divided also into two subsections

* + GRANTOR --> This provide privileges on the target objects to USER/SCHEMA so that DML operation can be executed on it.
  + GRANTEE --> This provide privileges on the objects of other schemas to target schema so that DML operation can be executed by it.

1. **Framework Execution**

## Master Job Execution

This master job has been created to run all the below functionality of the Environment Sync up framework. This job needs to execute in STAGE APP Box.

* Pentaho – trans and jobs folder sync up from source to target directory
* Temp\_Files – Copy files like TT files, Reference Look up files etc. from source to target folder.
* DB Objects – This take to sync up the objects like – Tables, Indexes, Store Procedure and Privileges.
* Purge Old files – This functionality remove 60 days old back up directories from Target & Temp\_Files folder as well as 30 days old folder from Metadata directory.
* Email Notification – Job sends email notification to support team in any case i.e. Success/Failure

Script Path: /paas/projects/envSetup

Command: nohup ksh environmentSetup.ksh “sourceSchema” “targetSchema”

Ex: nohup ksh environmentSetup.ksh “migration\_8” “migration\_7”

## Database Job Execution

This job has been created to run the database objects Environment Sync up like – Tables, Indexes, Store Procedure and Privileges. This job needs to execute in STAGE APP Box

Script Path: /paas/projects/envSetup

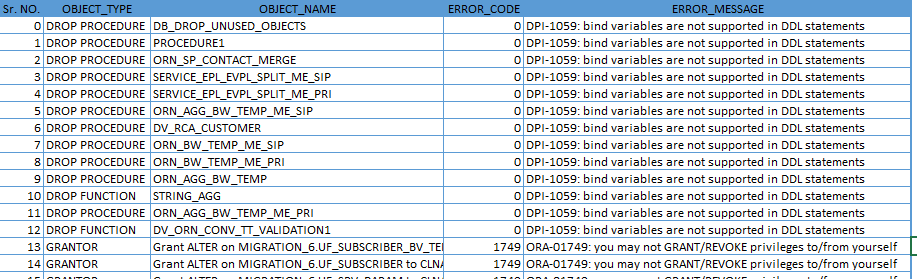
Command: ./environmentSetup.py “sourceSchema” “targetSchema” “metaDataPath”

Ex: ./ environmentSetup.py “migration\_8” “migration\_7” “migration\_5\_migration\_7\_0603”

1. **Database Error**

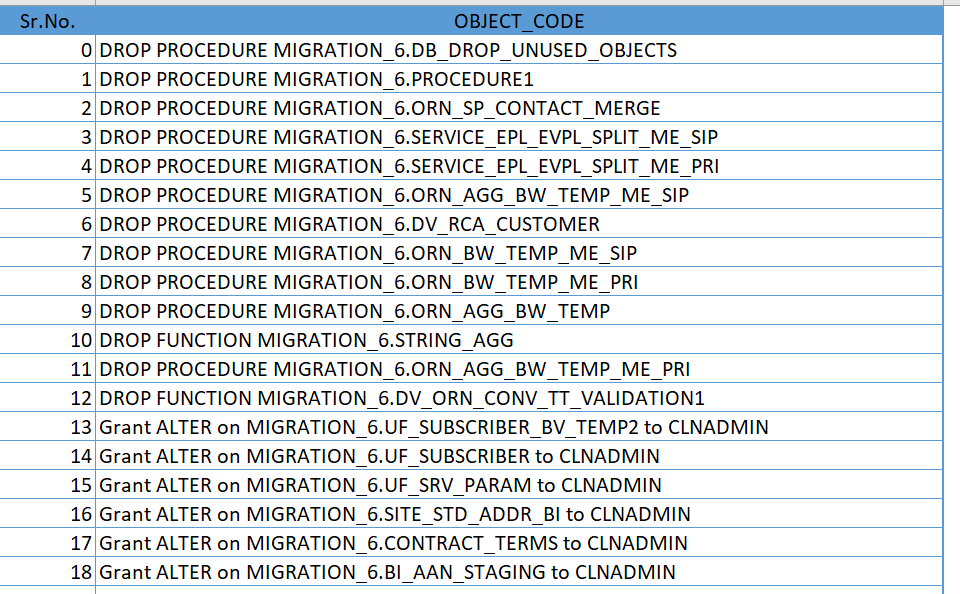
## Exception Details

There could be number of reasons that Database throws an error and framework intelligent enough to caught object’s error. The error details can be found “exceptionDF\_\*” file in Metadata folder with facts OBJECT\_TYPE, OBJECT\_NAME, ORACLE ERROR\_CODE and ORACLE ERROR\_MESSAGE



## Exception Objects DDL

The data definition of error object captured in “exceptionDDLCodeDF\_\*” file for further analysis.



1. **Related Documents**

|  |  |  |
| --- | --- | --- |
| **Document** | **File Name** | **Attach File** |
| Shell Script | environmentSetup.ksh |  |
| Python Script | environmentSetup.py |  |
|  |  |  |