|  |
| --- |
| Performance Contractors |
| ETL Overall Process |
| Enterprise Data Warehouse |

|  |
| --- |
| Indusa  12-19-2016 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision #** | **Date** | **Author** | **Notes** |
| 00 | 12/19/2016 | NT | Initial revision |
| 01 | 01/13/2017 | NT | Added the ETL flow Example packages |
| 02 | 01/24/2017 | NT |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

[Key Objective 2](#_Toc472095063)

[Environment Specifications 2](#_Toc472095064)

[Development 2](#_Toc472095065)

[UAT 2](#_Toc472095066)

[Production 2](#_Toc472095067)

[Database and Data Source Specifications 2](#_Toc472095068)

[AX & Legacy Source database 2](#_Toc472095069)

[Data Mart database 3](#_Toc472095070)

[Control Database 3](#_Toc472095071)

[ETL Process 3](#_Toc472095072)

[Extract 3](#_Toc472095073)

[Transform and Load 4](#_Toc472095074)

[Control Framework 6](#_Toc472095075)

[ETL Process for Dim Project: 11](#_Toc472095076)

[Matrix for the Control Fields Logic: 18](#_Toc472095077)

# Key Objective

The purpose of this document is to help user understand the ETL process, databases used for the Performance Contractor’s Enterprise Data Warehouse Solution developed by HITACHI and ETL Control Framework.

# Environment Specifications

Different environments used in the solution are as following.

## Development

The Development environment is used by developers to create/modify ETL SSIS Packages and SSRS reports. After the SSIS packages/ SSRS reports are created/modified, tested and approved then the changes are promoted to the UAT environment. The database server for Development is **BR42\DWSQL**.

## UAT

The UAT environment is used for testing by Business Analyst. If any SSIS package/SSRS report needs validation to be tested, then those verifications are done in UAT environment. The database server for UAT is **BR42\DWUAT**.

## Production

The database server for Production is **BR152\DWPRD**.

# Database and Data Source Specifications

The Different databases/data sources used in the solution are as following.

## AX & Legacy Source database

This the source database containing all the transactional data across the time about the organization. The source is different for each of the environment mentioned above

* 1. Development Source: BR42\DWSQL.PCI\_AX2012\_R2\_PRD
  2. UAT Source: BR42\DWUAT.PCI\_AX2012\_R2\_PRD
  3. Production Source: BR153\DWPRD.PCI\_AX2012\_R2\_PRD

The Staging database is the copy of the AX and Legacy Source database. This database is named as **PC\_EDWSTAGING** database. This name is common for all the three environments. The Staging database has three schemas with in it.

* 1. **AX Schema:** This schema is used to differentiate the AX staging table from Legacy staging table. The tables having AX as schema are the extracted copy of the AX Source tables.
  2. **STG Schema:** This schema is used for Legacy staging tables. The tables having the STG schema are the extracted copy of the Legacy tables.
  3. **Transform Schema:** The tables in this Schema are populated after the Transform packages in ETL executes. This tables are a bridge between the Staging tables & the DataMart.

## Data Mart database

The Data Mart database is the data warehouse created from the source. The data warehouse will contain Dimensions & Facts/Measures as data. The data warehouse is named as **PC\_EDWDATAMART** database. The DataMart consists of 2 schemas.

* 1. **DIM Schema:**  The DIM Schema contains all the Dimension tables in the data warehouse. The Dimension tables have the Detailed information for that Dimension.
  2. **FACT Schema:**  The FACT Schema contains the Fact tables that have Keys from Dimensions and the actual Measure values.

## Control Database

The Control database is used to monitor the Logs of the Packages, their execution, Error handling of SSIS ETL packages etc. It is named as **PC\_EDWCONTROL** database. The Control database consists of 3 schemas.

* + - 1. **Control Schema:** Control schema contains the information of the SQL Server agent Jobs that are executed from Control Framework. Various information regarding the Batch that executes the Job in **BatchLog**, **Runlog** of each Batch, Information regarding packages, **MetricLog** contains the information as in to how many rows were Inserted/Updated/Deleted in the packages, **MessageLog** table contains the SSIS error & Warning packages that it faces during the execution of package, **PackageLog** table contains the execution information of each package.
      2. **BI\_ACC Schema:**  The BI\_ACC Schema contains the information of the T-SQL Update Statements that are executed for Batch updates in SSIS packages from Control Framework.
      3. **Data Validation Schema:**  The Data Validation schema contains the information on Data Quality information for each package execution.

# ETL Process

ETL is the process of Extracting, Transforming and Loading the data from the Source into Data ware house.

## Extract

In this process, we create a copy of the source database. This copy is called the Staging database, and further the data manipulation is performed on this database. This step is performed using SSIS packages. The AX\_EXTRACT SSIS packages for AX Data Source and STG\_EXTRACT SSIS for Legacy Data Source. The AX\_EXTRACT packages are executed every night to get the latest transactional data from AX while, the STG\_EXTRACT packages for Legacy are one time loaded, as it won’t be having any new transactional data within it. The Staging database created is PC\_EDWSTAGING.

* 1. Here SCD Type -2 is used in creating the packages for extract packages, in this type all the historical data is stored in the destination table, the flag columns and date columns are used to differentiate the recent record from the historical records.
  2. For incremental Load packages (AX\_EXTRACT\_I\_), the incremental load is implemented, i.e. if the record is already there then no need to change the records else, it will only insert the records in the destination table.

**PC\_EDWSTAGING**

Extract (AX\_EXTRACT & STG\_EXTRACT SSIS packages)

**AX**

**Legacy**

Source data

Fig.1 Extract

Fig.1 Extract

## Transform and Load

Here, the data is cleaned, and transformed into a common format to then the data is loaded in Data ware house. Here the data manipulation is performed on **PC\_EDWSTAGING**. The data is then Loaded in **PC\_EDWDATAMART** database.

* 1. For late arriving dimensions, we first create the ID in the Dimension if it does not exist in Dimension table and set the column for the flag column **LateArrivingFlag**, then whenever the values for that records arrive, the package will update the record and reset the value for the flag column **LateArrivingFlag**

**PC\_EDWSTAGING**

Transform (AX\_TRANSFORM\_ SSIS packages)

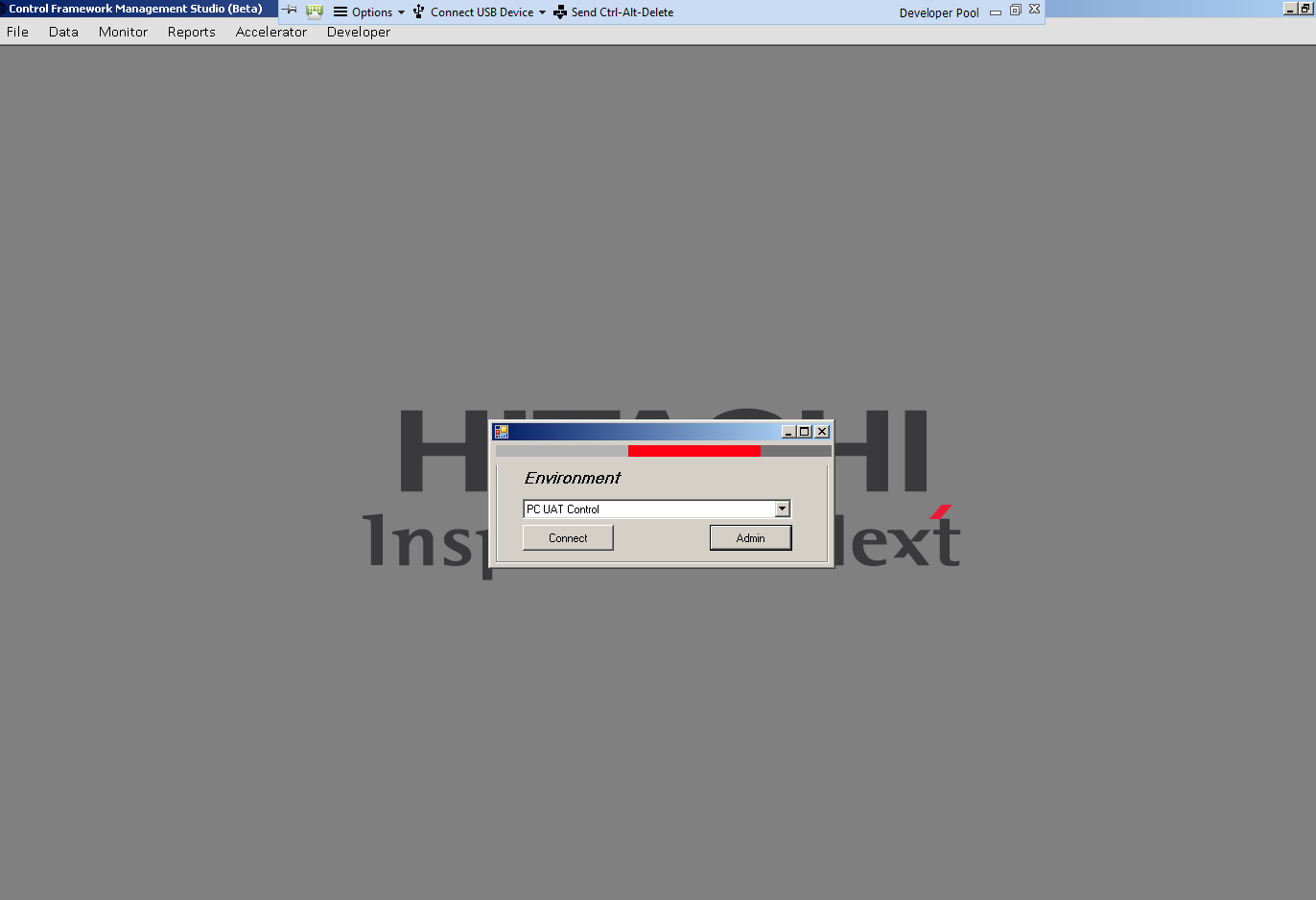
Load (AX\_LOAD\_ SSIS packages)

**PC\_EDWDATAMART**

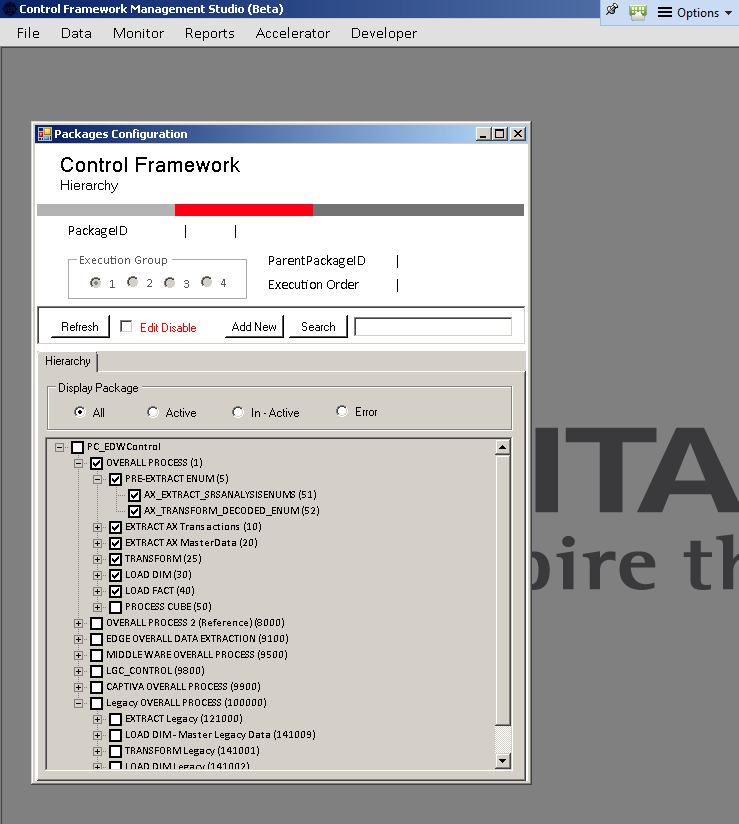
Fig.2 Transform & Load

# Control Framework

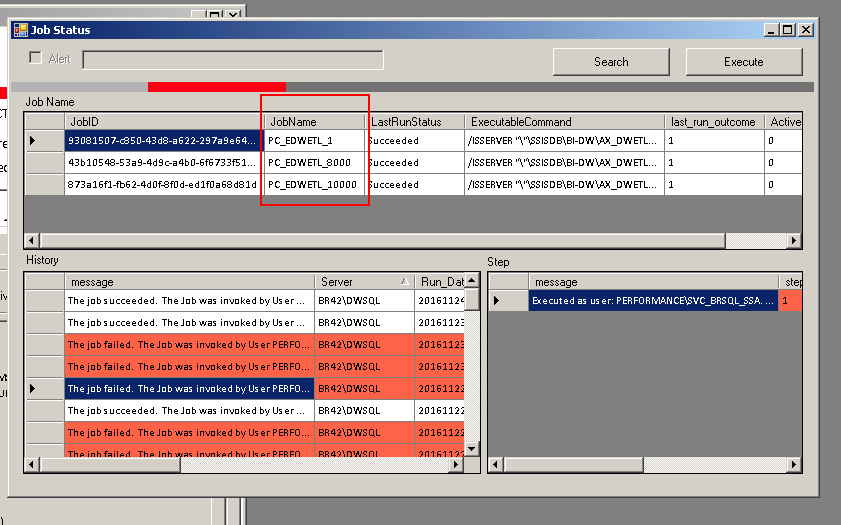
Control framework is used for monitoring the ETL process across different environments; the framework executes the SQL Server agent jobs. There are different group of hierarchies for packages in framework.



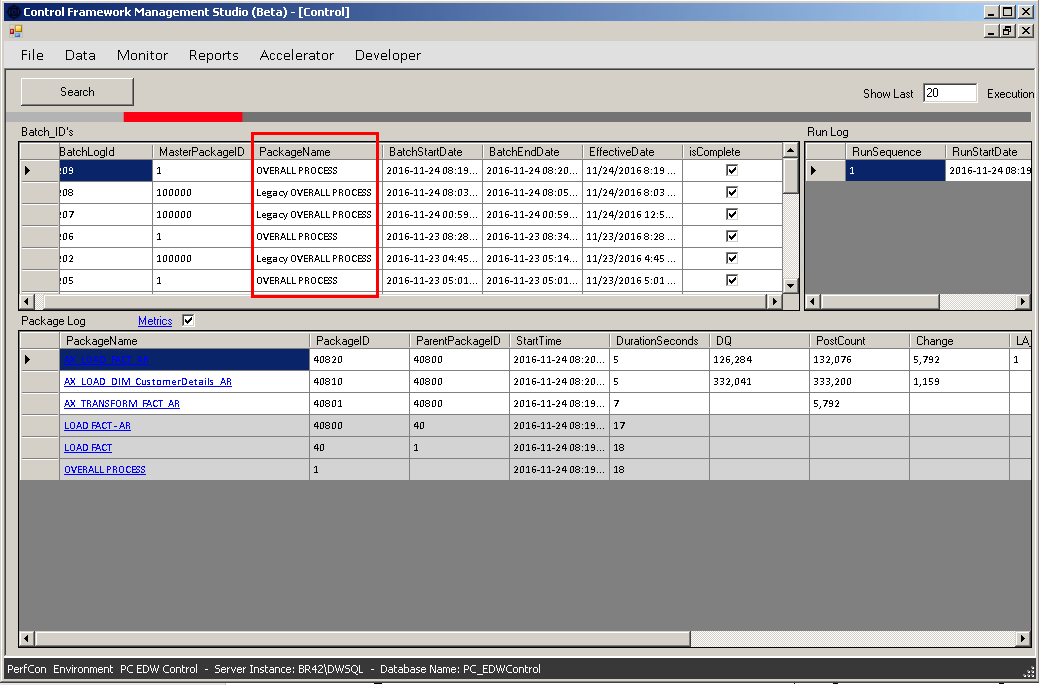
The Hierarchy tab is used to select and deselect the packages for execution. User can select/deselect the packages from **DATA -> Hierarchy**



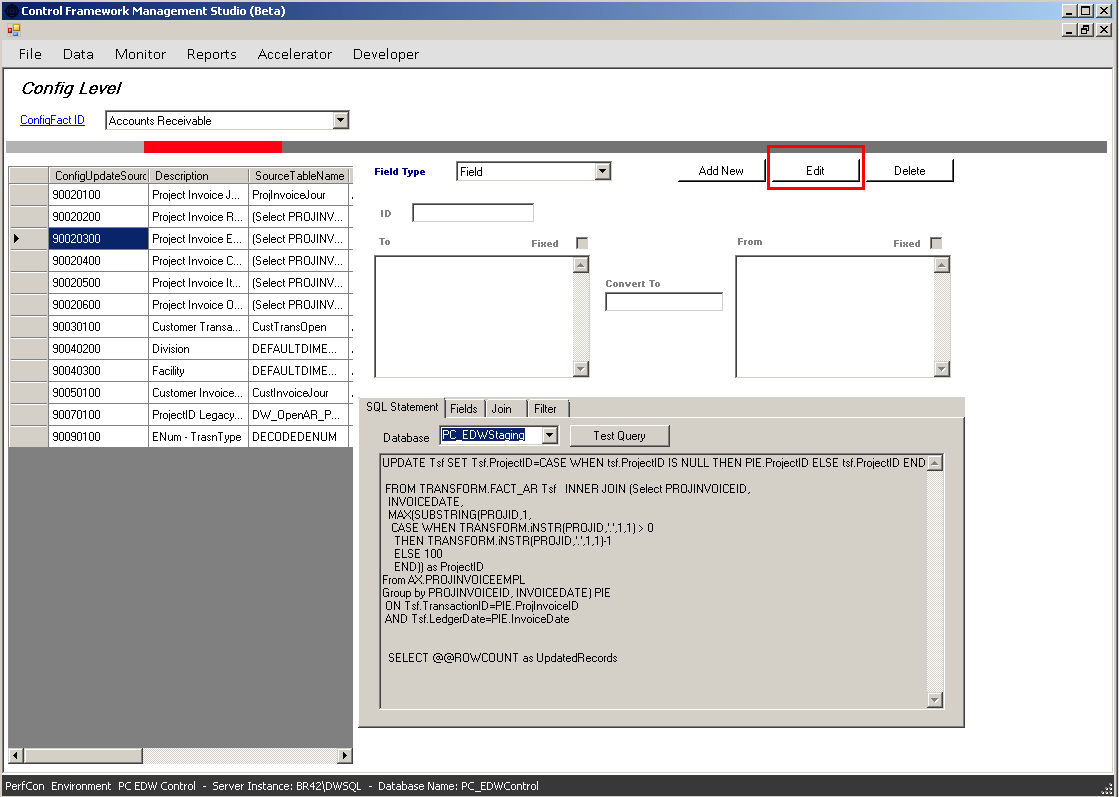
In the control framework, The **Job Status** menu in **Monitor** tab, executes the SQL server agent job which execute the SSIS packages as Job Steps,



User can check the Status of the Package execution through Job from **Monitor -> Control**.



User can change the update Query used in the SSIS package tasks from **ACCELERATOR -> DATA TRANSFORMATION -> FIELDS METADATA**



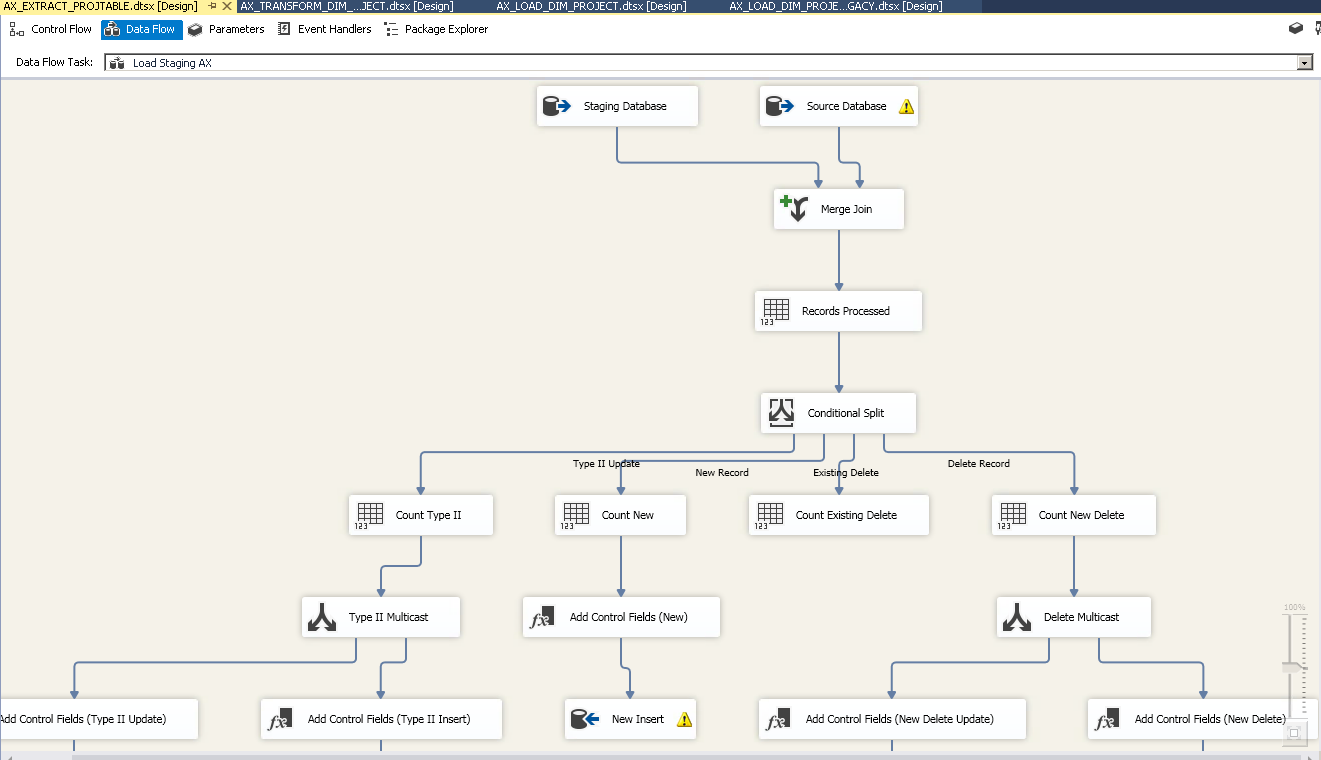
**PC\_EDWCONTROL** database has the information related to the SSIS packages that are executed using the SQL Server Agent jobs executed from framework. When we make an entry in **PC\_EDWControl** database (table: **CONTROL.PACKAGE**), then, there will be a **corresponding** entry in **Control framework**.

**NOTE: Whenever you create an SSIS package you must make an entry in CONTROL.PACKAGE table, so that you can see the newly created Package in Control Framework Hierarchy.**

# ETL Process for Dim Project

**Extract package: AX\_EXTRACT\_PROJTABLE package**

This package will create a copy of the PCI\_AX2012\_R2\_PRD.dbo.PROJTABLE to PC\_EDWStaging.AX.PROJTABLE. The PROJTABLE table is Master table that has the actual master data as information



All the package that are used to fill the master data are having the SCD Type 2 i.e. Slowly Changing Dimension Type 2 changes implemented. The data is first merged join with the destination to check if the record already exists or not.

The MD-5 Algorithm is used to do the hashing and get the encrypted value for each record.

After merging the records in Source and destination using the full outer join the records are processed and moved to Conditional split, where the records are checked and classified as they are new records, SCD Type changed records or Deleted records,

The conditions are classified here are:

1. **New Record:** The SSIS Expression for this condition is **ISNULL(STG\_HTotal)**, which means that the record exists in the source, but not in Staging Destination table, so the record must be inserted.
2. **Delete Record:** The SSIS Expression for this condition is **ISNULL(SRC\_HTotal) && STG\_DeletedFlag == FALSE**, this expression means that the records is not present in source table, but it is present in Destination table, also the DeletedFlag column, which is a control field used to differentiate current records from deleted records, is set to false.
3. **Existing Delete:** the SSIS Expression for this condition is **ISNULL(SRC\_HTotal) && STG\_DeletedFlag == TRUE**, this means that the record once existed in the source table and then the record is Logically deleted from the Destination
4. **Type II Update:** The SSIS Expression for this condition is **SRC\_HTotal != STG\_HTotal || STG\_DeletedFlag == TRUE**, which means that the record does exists in the Destination table, but there are some data changes in the records, also the record is marked as the Stale version using the DeletedFlag.

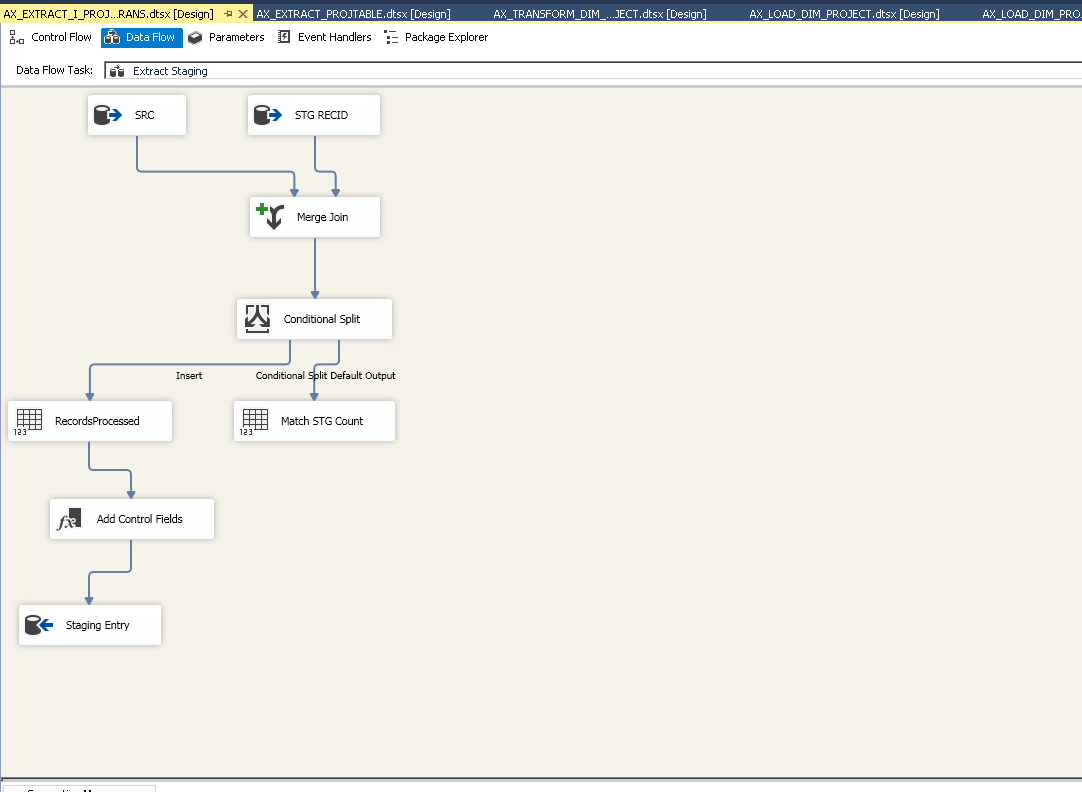
All these condition perform some or the other operation on the record.

1. **New Record:** The record will be inserted to the destination table
2. **Delete Record:** Here the record existing in Destination will be marked as Deleted using the T-SQL update, the CurrentFlag will be set to False and the DeletedFlag is already set to False, the records having the CurrentFlag and DeletedFlag as false means that they are the Stale versions of the records that once existed in Source. A new record will be inserted in the destination table with the CurrentFlag=1 and DeletedFag=1, this means that it is the Current version of the Deleted records.
3. **Existing Delete:** No changes are done for this condition as the record is already marked as deleted from Destination table. The records here will only be counted and stored to a variable.
4. **Type II Update:** Here the operation performed will be similar to the Delete Records, the records existing in the Destination will be updated, the Current Flag will be Set to False and Deleted Flag will be set True as the record is now logically deleted from the Destination table. A new record will be inserted with CurrentFlag set to True and DeletedFlag set to False.

**NOTE: No record will be physically deleted for Master data. It will be marked as the Deleted using the Control Fields.**

**Extract package: AX\_EXTRACT\_I\_PROJEMPLTRANS**

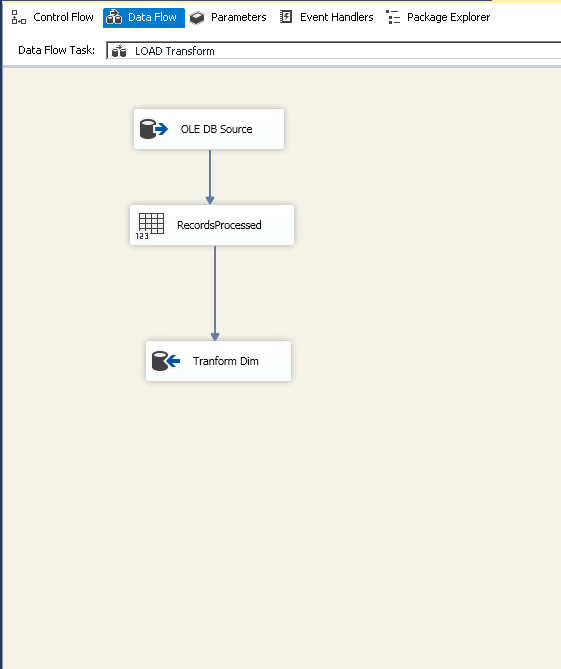
This package will create a copy of PCI\_AX2012\_R2\_PRD.dbo.PROJEMPLTRANS table to PC\_EDWStaging.AX.PROJEMPLTRANS table. PROJEMPLTRANS is the transactional table, so it will have a different development.



For the Transactional data only one condition is checked that whether the transactional record exists in the destination or not, If the record doesn’t exist in the destination then it will be inserted, else no operation will be performed

**Transform package: AX\_TRANSFORM\_DIM\_PROJECT**

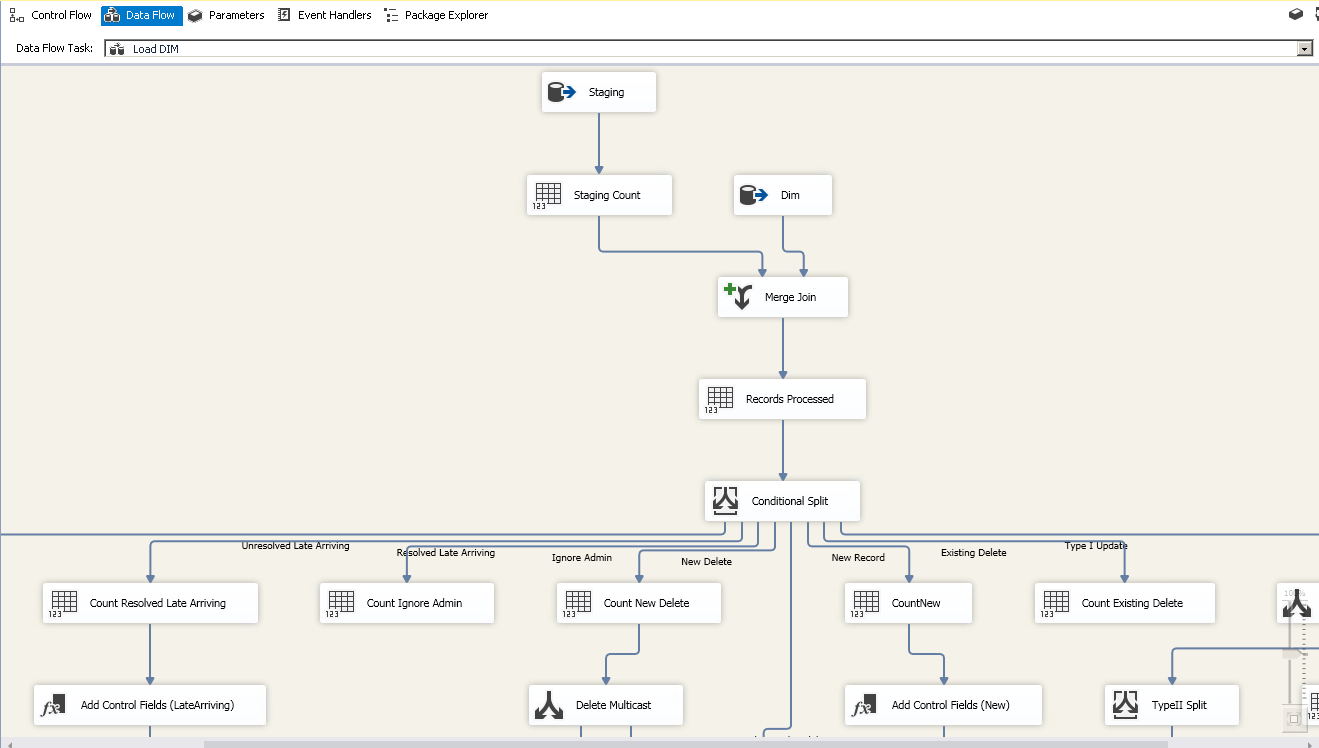
This package will transform the records will be inserted in Transform.DIM\_PROJECT.



No operations are performed on transform packages.

**Load package: AX\_LOAD\_DIM\_PROJECT**

The manipulated and transformed data will insert to PC\_EDWDataMart.DIM.PROJECT table. Since it is a Dimension, it will have Master data for that dimension.



All the package that are used to fill the master data for Dimensions are having the SCD Type 1 i.e. Slowly Changing Dimension Type 1 changes implemented for updates in the records. The data is first merged with the destination to check if the record already exists or not.

The MD-5 Algorithm is used to do the hashing and get the encrypted value for each record. Two hash columns are used **HTotalTypeI** and **HTotalTypeII**, the former is used for **SCD Type 1** update, while the latter is used for **SCD Type 2** updates for logically deleted records. For **HTotalTypeI** column hashing is done for all the columns, for **HTotalTypeII** only a blank value is used to get the values.

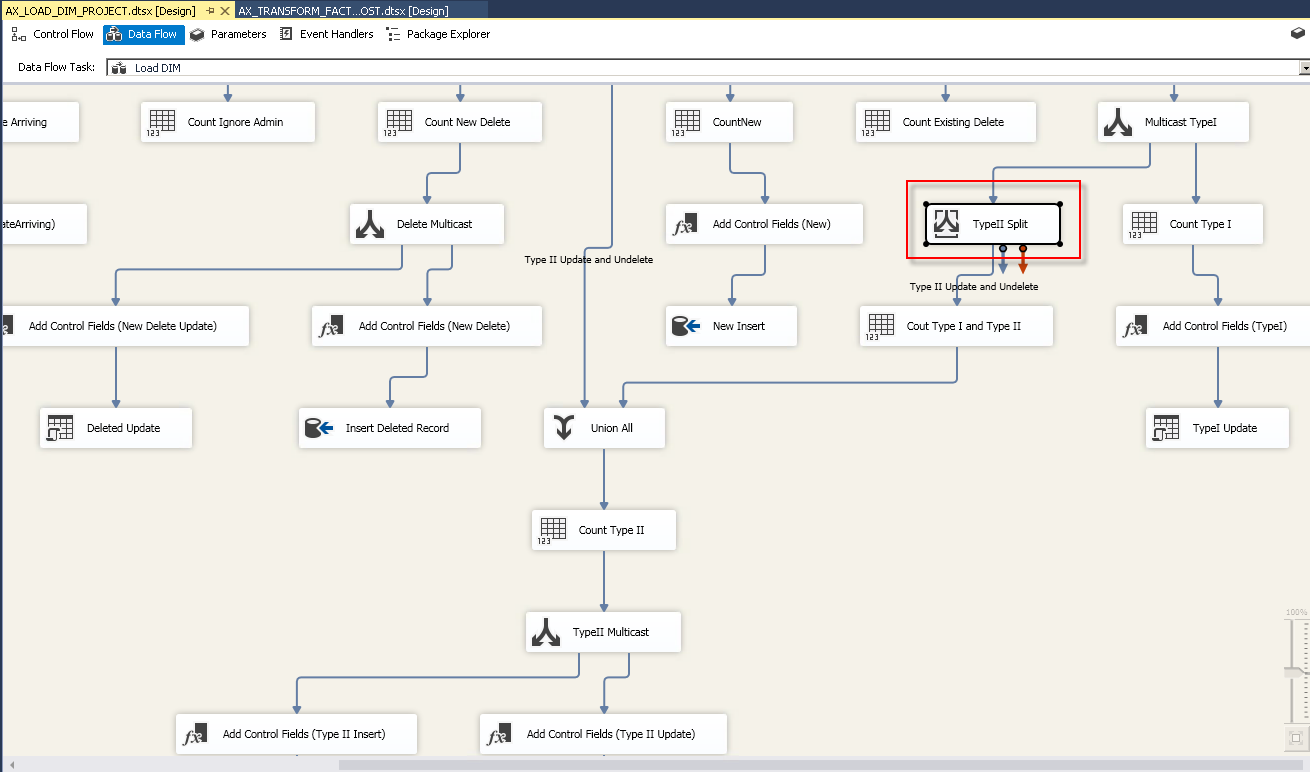
After merging the records in Source and destination using the full outer join the records are processed and moved to Conditional split, where the records are checked and classified as they are new records, SCD Type changed records or Deleted records,

The conditions are classified here are:

1. **New Record:** The SSIS Expression for this condition is **ISNULL(DW\_HTotalTypeII)**, which means that the record exists in the source, but not in Staging Destination table, so the record must be inserted.
2. **New Delete:** The SSIS Expression for this condition is **ISNULL(STG\_HTotalTypeII) && DW\_DeletedFlag == FALSE,** this expression means that the records is not present in source table, but it is present in Destination table, also the DeletedFlag column, which is a control field used to differentiate current records from deleted records, is set to false.
3. **Existing Delete:** the SSIS Expression for this condition is **ISNULL(STG\_HTotalTypeII) && DW\_DeletedFlag == TRUE,** this means that the record once existed in the source table and then the record is Logically deleted from the Destination
4. **Type II Update and Undelete:** The SSIS Expression for this condition is **STG\_HTotalTypeII != DW\_HTotalTypeII || DW\_DeletedFlag == TRUE,** which means that the record does exists in the Destination table, but there are some data changes in the records, also the record is marked as the Stale version using the DeletedFlag.
5. **Ignore Admin:** The SSIS Expression for this condition is **ISNULL(DW\_HTotalTypeII)**.
6. **Unresolved Late Arriving Flag:** The SSIS expression for this condition is **ISNULL(STG\_HTotalTypeII) && DW\_LateArrivingFlag == TRUE**, this means that the records is inserted in any of the FACT tables, but the corresponding record entry is not found in Dimension table.
7. **Resolved Late Arriving Flag:** The SSIS expression for this condition is **!ISNULL(STG\_HTotalTypeII) && DW\_LateArrivingFlag == TRUE**, this means that the record is now present in Source , so we can fill the corresponding records in Dimension
8. **Type I Update:** The SSIS expression for this condition is **STG\_HTotalTypeI != DW\_HTotalTypeI**, this means that the record does exists in the dimension table, but some of the data is updated in Source. After checking the **Type I Update** condition, another condition is also checked for **Type II Update and Undelete**, the expression for this is **(STG\_HTotalTypeII != DW\_HTotalTypeII) || (DW\_DeletedFlag == (DT\_BOOL)1).**

All these condition perform some or the other operation on the record.

1. **New Record:** The record will be inserted to the destination table
2. **Delete Record:** Here the record existing in Destination will be marked as Deleted using the T-SQL update, the CurrentFlag will be set to False and the DeletedFlag is already set to False, the records having the CurrentFlag and DeletedFlag as false means that they are the Stale versions of the records that once existed in Source. A new record will be inserted in the destination table with the CurrentFlag=1 and DeletedFag=1, this means that it is the Current version of the Deleted records.
3. **Existing Delete:** No changes are done for this condition as the record is already marked as deleted from Destination table. The records here will only be counted and stored to a variable.
4. **Type II Update:** Here the operation performed will be similar to the Delete Records, the records existing in the Destination will be updated, the Current Flag will be Set to False and Deleted Flag will be set True as the record is now logically deleted from the Destination table. A new record will be inserted with CurrentFlag set to True and DeletedFlag set to False.
5. **Ignore Admin:** No operations are performed for this condition
6. **Unresolved Late Arriving Flag:** No operations are performed for this condition
7. **Resolved Late Arriving Flag:** The record with LateArrivingFlag set to True is updated and the Flag is reset.
8. **Type I Update:** The record is updated in Destination table. If a record is having any updates in data, then all the records will be having the same business keys will be updated, and also the logic for SCD Type-2 will be checked to know if the records have changes in the attributes that needs to maintain historical data or not. Currently, only blank is used to verify the SCD Type 2 updates.



**NOTE: To know if the attributes in Dimension are having SCD Type1 or SCD Type 2 updates, user can check the BI Role column in Dimension Design Documents in TECHDOCS folder**

# Matrix for the Control Fields Logic:

|  |  |  |
| --- | --- | --- |
| **CurrentFlag** | **DeletedFlag** | **Implication** |
| 1 | 0 | Current record |
| 0 | 1 | Type -2 historical record |
| 1 | 1 | Current version of the record not present in Source (deleted from the source) |
| 0 | 0 | No longer valid as not present in the Source |

**NOTE: No record will be physically deleted from the Data warehouse. It will be marked as the Deleted using the Control Fields.**