

**RMIT Project Pi**

# Data Migration Validation Solution

Author: Deloitte

Date: 01 October 2019

Version: 0.1

**RMIT Project Pi**

# Data Conversion Scope and Strategy

Author: Deloitte

Date: 28 May 2019

Version: 0.10

**CONFIDENTIAL – NO DECISION HAS BEEN MA**

**Document Control Information**

**Document Information**

|  |  |
| --- | --- |
| Deliverable Identification | CNV 150 |
| Document Name | Data Validation Solution |
| Project Name | RMIT Project Pi |
| Client | RMIT |
| Document Author |  |
| Document Version | 0.1 |
| Document Status | In Progress |
| Date Released |  |

**Document Edit History**

| Version | Date | Additions/Modifications | Prepared/Revised by |
| --- | --- | --- | --- |
| 0.1 | 01/10/2019 | Creation of Document | Data Conversion project team |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Document Review**

| Date | Name | Organization/Title | Comments |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Approval History**

The following people are designated recipients of the final version of this document:

| Date | Name | Organization/Title |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Table of contentS**

[1. Introduction 4](#_Toc23362954)

[2. Dependencies, Risks and Assumptions 7](#_Toc23362955)

[**2.1.** **Dependencies** 7](#_Toc23362956)

[**2.2.** **Risks** 7](#_Toc23362957)

[**2.3.** **Assumptions** 7](#_Toc23362958)

[3. ETL Architecture 8](#_Toc23362959)

[4. Technical validation solution 9](#_Toc23362960)

[5. Data Validation Framework 13](#_Toc23362961)

[5.1. QAQC Framework 14](#_Toc23362964)

[5.1.1. QAQC\_CATEGORY 15](#_Toc23362965)

[5.1.2. QAQC\_RULES 15](#_Toc23362966)

[5.1.3. QAQC\_RULE\_BOOK 16](#_Toc23362967)

[5.1.4. QAQC\_RULE\_CHECK\_RESULT 16](#_Toc23362968)

[5.1.5. QAQC\_FUNCTION 17](#_Toc23362969)

[5.1.6. UPDATE\_DQ\_CHECK\_RESULT 17](#_Toc23362970)

[5.2. Batch control Framework 17](#_Toc23362971)

[5.2.1. ETLLoadInfo 17](#_Toc23362972)

[5.2.2. ETLLoadSummary 18](#_Toc23362973)

[5.2.3. ETL\_Execution\_Dependency 18](#_Toc23362974)

[5.2.4. Row\_Count\_Info 19](#_Toc23362975)

[5.2.5. INSERT\_AUDIT\_INFO Procedure 19](#_Toc23362976)

[5.2.6. UPDATE\_AUDIT\_INFO 19](#_Toc23362977)

[5.2.7. INSERT\_ROW\_COUNT\_INFO 19](#_Toc23362978)

[5.3. Exception handling framework 20](#_Toc23362979)

[5.3.1. AUDIT\_ERRORLOG 20](#_Toc23362980)

[5.3.2. Identifying Exception 20](#_Toc23362981)

[5.3.3. Exception Classification and Resolution plan 21](#_Toc23362982)

[5.3.4. Exception Correction and Reprocessing 22](#_Toc23362983)

[6. Technical Validation Reports 24](#_Toc23362984)

[7. Appendix 27](#_Toc23362985)

[7.1. References 27](#_Toc23362986)

[7.2. QAQC Entity column definitions 27](#_Toc23362987)

[7.2.1. DQ.CATEGORY 27](#_Toc23362991)

[7.2.2. DQ.RULES 27](#_Toc23362992)

[7.2.3. DQ.RULE\_BOOK 28](#_Toc23362993)

[7.2.4. DQ. RULE\_CHECK\_RESULT 29](#_Toc23362994)

[7.2.5. DQ.FUNCTION 30](#_Toc23362995)

1. Introduction

RMIT operates in a changing and complex sector that is evolving rapidly. RMIT’s 2020 Strategy, Ready for Life and Work, sets out a strategic response to ensure greater operational effectiveness through clearer, smarter and simpler systems. To deliver on this aspiration RMIT must overcome a number of critical existing pain-points with the processes and systems used across Human Resources (HR), Financial Services, Procurement and the Student Information System (Student) that each play an integrated role in RMIT’s day-to-day business operations. Project Pi is going to improve the staff and student experience at RMIT by unifying our processes and introduce Workday as our new Enterprise Resource Platform (ERP) for all of the University’s HR, Finance and Procurement functions. The project is a true business transformation program, involving redesigning core business processes, and embracing contemporary ways of working and new technology.

This document presents a high-level data migration validation solution on data conversion process. This strategy provides RMIT with a proven approach towards data validation based on best practices and Deloitte’s past implementation experience

* 1. Document Purpose

The purpose of this document is to define the solution to execute technical validation procedures to validate data migrated from legacy applications to target applications.

The Data Validation solution includes:

* Identification of source and target systems for carrying out validation
* Design of the overall process for validating the data throughout the data migration cycle
* Highlighting assumptions, risks and dependencies
* Design the data model and ETL Framework to capture, persist & report data issues in relation to data migration
* Details of the data validation activities performed across pre-load, transform and post-load stages (e.g. technical reconciliation and data validation)

This document acts as a guide for the data conversion and RMIT functional teams, and will act as:

* A working document for validating data across multiple stages of migration

A reference guide for stakeholders and relevant program teams

* 1. Key Document Terms

The following table outlines key terms which are used throughout the data migration technical validation document.

| **Abbreviation/Name** | **Full Name/Definition** |
| --- | --- |
| CSV | Comma-Separated Values |
| ETL | Extract, Transform, Load |
| Functional Team | Includes Deloitte Workday HR, Deloitte Workday Finance and RMIT Business |
| Hoover | Deloitte Proprietary Tool |
| SIT | System Integration Testing |
| SSIS | SQL Server Integration Services |
| Workday | Financial management and human capital management software vendor |
| BCF | Batch Control Framework |
| QCF | Quality check Framework |
| RCA | Root Cause Analysis |
| KPI | Key Performance Indicators |
| H1,H2,H3 | Hop 1, Hop2 , Hop3 |
| DQ | Data Quality |

* 1. Audience

The target audience for this document is:

* Data Conversion project team
* HR Lead
* HR Project Manager
* Finance Lead
* Finance Project Manager
* Procurement Lead
* Procurement Project Manager
* Technology Lead
* Technology Project Manager
* RMIT Operations team

* 1. Scope

The validation solution scope is based on the configuration elements defined within the target applications, which include HR, Finance, Payroll and S2C/Contract Lifecycle Management. The solution will cover validation of all data records required to support the end state of the target solutions. The Data Conversion project team will execute the data validation activities for the end-to-end ETL. This will include validating data between source systems and landing area, landing area to staging area and staging area to target system.

The data validation requirements for non-Workday systems like ADP and Icertis are expected to align with the proposed data validation solution. The Data Conversion project team will work with the non-Workday system conversion leads to align on activities and timelines, and agree on dependencies.

* 1. Out of scope

All activities performed by RMIT business to reconcile data between source and target systems are out of scope for this solution. The details of business reconciliation are covered in CNV130 Data Migration Reconciliation Solution.

1. **Dependencies, Risks and Assumptions**
   1. **Dependencies**

The data conversion process relies on the completion of certain supporting activities, which are outlined below.

| **No.** | **Description** |
| --- | --- |
| 1 | Tools, System and application specific access will be owned by RMIT and will be made available to teams as per policy |
| 2 | Business team representative or Interfacing teams are responsible for providing data in the landing area as per timelines of the project |

* 1. **Risks**

All the risks and mitigation strategies are covered as part of Pi\_DTA\_Data\_conversion\_strategy document.

| **No.** | **Description** |
| --- | --- |
| 1 | Any issue that may restrict the custom solution to use SQL Server bulk utility to load the workday extract file data into SQL Server would require an alternate approach and hence delay the timelines to fully automate technical validation solution |

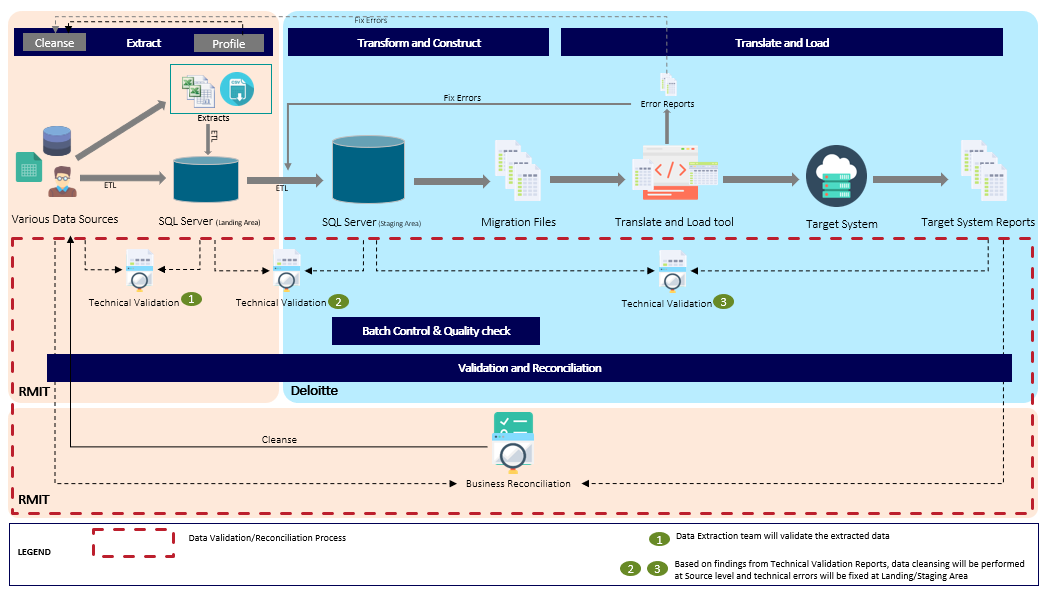
* 1. **Assumptions**

The contents of this document have been compiled based on the following assumptions related to the data conversion process.

| **No.** | **Description** |
| --- | --- |
| 1 | All necessary data is available in the extracts finalized as part of scope. Any new feeds realized during implementation phase will have to be evaluated and estimated for |
| 2 | Business team representative or Interfacing teams will be able to provide data in landing area as per timelines of the project that will be created |
| 3 | Additional data clean-up or standardization will be done by RMIT business |
| 4 | All required RMIT resources are available to perform required work, attend meetings, review and provide approvals during project timeframe. |
| 5 | All deliverables developed by Deloitte Consulting will be in English only |
| 6 | RMIT is responsible for providing facilities for the engagement team, including sufficient work space, system and network access, and phone access. |

1. ETL Architecture

RMIT future state architecture is composed of various components including the Batch Control and Quality Check frameworks. The following diagram outlines the overall solution architecture.



As part of this document Batch Control, Quality check frameworks and Technical validation solution will be explained in subsequent sections. For more details on overall architecture please refer to Pi\_DTA\_Data\_conversion\_strategy document

1. Technical validation solution

Technical validation of data is required at different stages of the conversion process as a control mechanism for each point of change or movement of information to ensure correctness, validity and quality of the data. Technical validations are typically programmed and automated (where feasible) using ETL tools and scripts. The criteria for technical validations are typically quantitative (record counts, load statistics) and are performed comprehensively across all data objects in-scope in the ETL cycles. Segregated across 3 hops(H1,H2 and H3), the data validation activities are carried out by different teams as showcased below in the high-level data conversion flow.



Validating the accuracy and usefulness of the converted data requires collaboration between both the technical and the functional teams. The Data Conversion project team is responsible for monitoring the accuracy and integrity of data at each point in the conversion process. Technical validation reports will be produced outlining any data that was dropped or unable to be loaded and also be used as input into data cleansing activities needed in source or transformation process:

The below diagram depicts the proposed validation steps



Technical validation will be performed at the following checkpoint

* ***Technical Validation 1(Hop 1): Source to Landing database***

The data in the source systems will be validated against extracted data in the Landing database. The validations include but not limited to control totals, row counts, financial metrics, duplicate records, field value comparison etc. Any data issues will be addressed by cleansing the data in source systems.

Technical validation 1 is not covered as part of this document and is owned by RMIT

* ***Technical Validation 2(Hop 2): Landing database to the Staging database***

The extracted data in the Landing database will be validated against the transformed data in the Staging database. The validations include but not limited to row counts, field value comparison in case of one-to-one mapping etc. Any data issues will be addressed by cleansing the data in source systems/landing database.

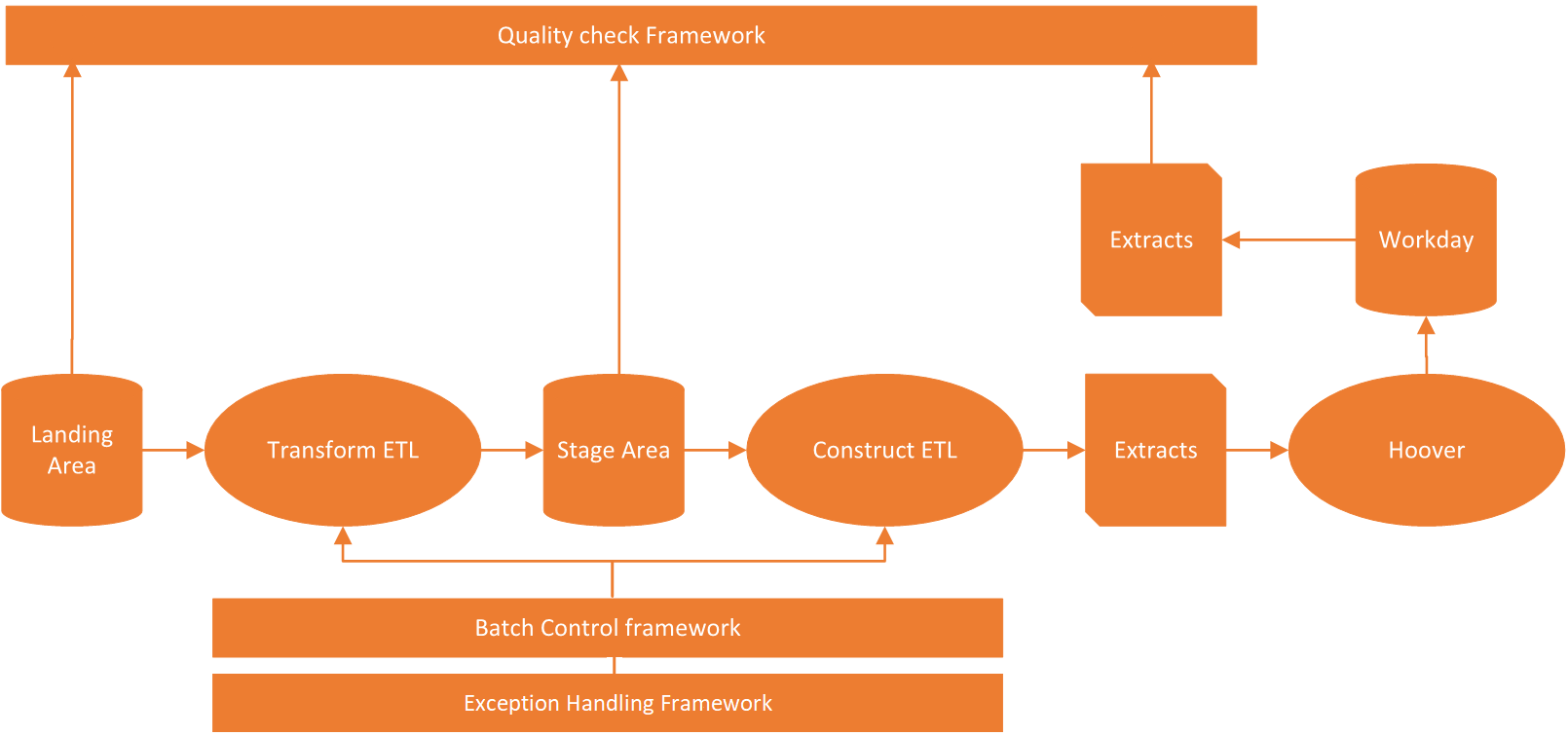
* **Validate Transformation rules**– This check involves validation of transformations applied to source field values against the target field values using the QCF framework. SQL query will be developed to transform data from landing area and compare with data from stage tables. These SQL queries will be populated in QCF static configuration table for consumption by the QCF ETL jobs which would perform source and target data validation and populate the result into QCF Result table.
  + **Validate Data Filters**– This check involves validation of filter criteria used in the source queries that restricts or aggregates the raw data. This check will be done manually during the Peer review process . All issues identified here will be fixed by updating the mapping specification or ETL code.
  + **Validate counts** – This check involves validating the source data count and target data count using the QCF framework. SQL query will be developed to fetch data count from landing area and staging area. These SQL queries will be populated in QCF static configuration table for consumption by the QCF ETL jobs which would perform landing database to staging database count validation and populate the result into QCF Result table.
  + **Validate duplicate data** – This check involves validating the uniqueness criteria defined in the file scope for each extracts. A logic will be developed to generate alternate keys in transform SSIS ETL and identify duplicates records based on the alternate key. All duplicate records will be captured through Batch control framework and redirected into error table.
  + **Validate Required field checks**– This check involves validation of required field values in target using the QCF framework. SQL query will be developed to check whether target data value is NULL or blank. These SQL queries will be populated in QCF static configuration table for consumption by the QCF ETL jobs which would perform source and target data validation and populate the result into QCF Result table.
* ***Technical Validation 3 (Hop 3): Staging database to target system reports***

The transformed data in the Staging database will be validated against the extracts generated from workday system. These workday extracts would be generated manually by hoover team and uploaded to secured location. Extract files will be loaded into staging area using custom solution and then the extract data will be compared against the stage table data using QAQC ETL Jobs. The validations include row counts and field value comparison. Any data issues will be addressed by either cleansing the data in source systems/landing database/staging database or updating translate and load tool scripts. Validation Step 7 and 8 are performed during this phase.

* + **Validate data issues** (Staging vs workday extracts)– This check involves validating counts (where applicable\*) and field values between source (stage layer) and target (workday reports). SQL query would be developed to fetch data from staging area and the workday extract file (the temporary table having the workday extract data populated through the custom solution). These SQL queries will be populated in QAQC static configuration table for consumption by the QAQC ETL job which would perform staging database vs workday extract validation and populate the result into QAQC Result table.
  + **Generate reports** : This step involves manually generating reports out of error table and QAQC check result table as per the reporting framework
* ***~~Technical Validation Shared Tasks:~~*** ~~The Validation framework compares data at different stages to ensure data is converted and loaded per mapping specification. Quality Assurance and Quality Control (QAQC) framework is built into the overall validation framework and is executed through an ETL job.~~ ~~Validation Step 1, 9 and 10 are performed during all validation phases.~~
  + ~~Step 1: Generate Audit Logs– This involves logging the validation execution details at runtime in the audit tables using the batch control framework. Execution state, runtime ETL failures and overall execution metadata is captured by the batch control framework during this step.~~
  + ~~Step 9: Investigate Error Records- This involves manually analyzing the validation result.~~
  + ~~Step 10: Generate Report- This step involves manually generating reports out of error table and QAQC check result table as per the reporting framework.~~

1. Data Validation Framework

Data validation framework consists of batch control, exception handling & QCF Framework which are integrated with the data migration ETL packages to log the metadata information in and perform automated validations. Below diagram describes the interdependencies between these frameworks at high level.



* Transform and Construct ETL packages leverage the batch control and exception handling framework to log the metadata information into audit and error table for every ETL executions. Audit and exception data will be used to monitor the data migration progress and generate exception reports. Please refer section 5.2 & 5.3 for technical details of Batch control framework.
* QCF framework validates the data loaded in the workday and staging area against the raw data. Please refer section 5 for technical details of QCF framework.

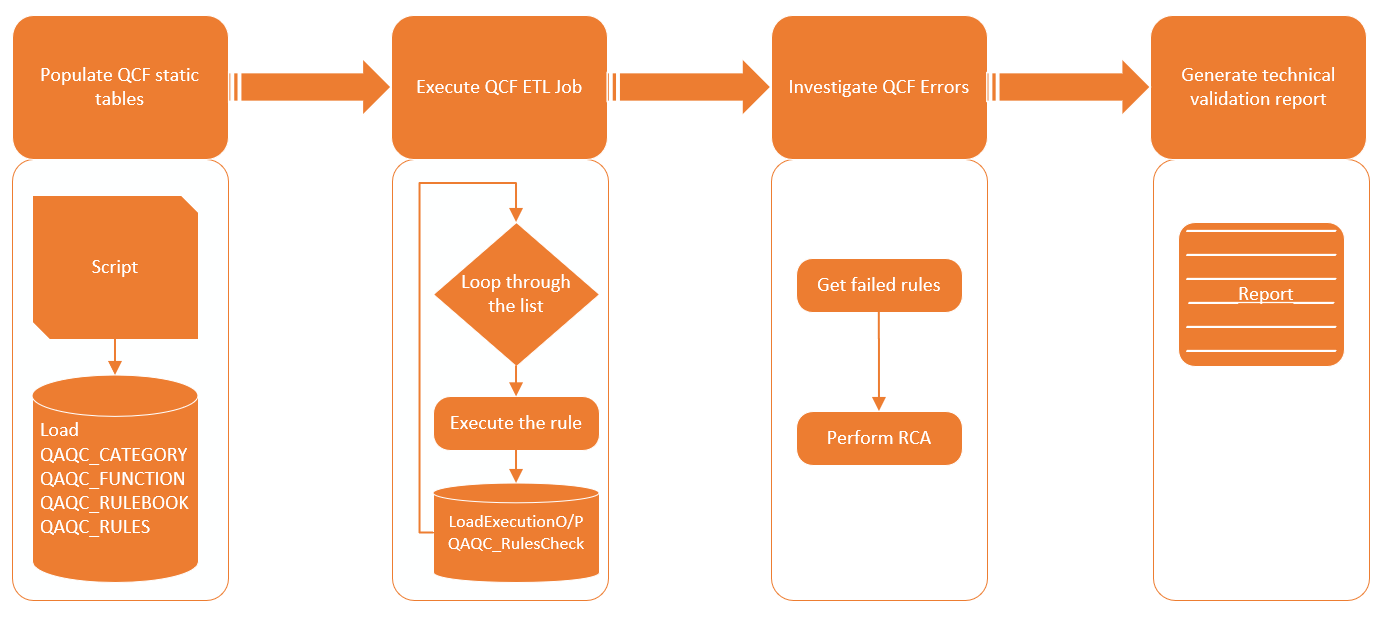
Attached below is the logical data model which contains all tables used as part of batch control framework, exception framework & QCF framework and interdependencies among them.



* **Batch control framework**: Batch control framework consists of set of 4 Audit tables and 3 stored procedures that manipulates data in the audit tables. All audit tables except ETLExecutionDependency are populated dynamically at runtime through the stored procedures. The ETLExecutionDependency consists of static master data.
* **Exception framework**: As part of exception handling process error records will be inserted into error tables, depending on business rules and quality checks. This framework consists of 1 error table which is populated dynamically at runtime through the SSIS ETL packages.
* **QCF framework**: QCF framework consists of set of 5 tables and 1 stored procedure that manipulates data in the QCF result table. All QCF tables except Rule\_Check\_Result consists of static configuration data. The Rule\_Check\_Result table is populated dynamically at runtime through the QCF ETL package.

2. 1. QCF Framework

An efficient, flexible, scalable and reusable data quality framework is very important in today’s world which can measure data quality with minimal cost and time as well as provide accurate results. Below diagram describes the tasks performed during QCF validation.



* Populate QCF static tables: This is one-time manual activity which involves populating the static configuration data in QCF tables.
* Execute QCF ETL Job: QCF ETL performs the technical validation based on the configured data. The ETL creates dynamic SQL query to subtract the target data from source data (Except operation) and identifies the data mismatches. The ETL loads the output of except operation into result table for analysis and reporting.
* Analyse the QCF ETL Result: This step involves analysing the result output and manually performing root cause analysis.
* Generate technical validation report: This step involves manually generating reports out of error table and QCF check result table as per the reporting wireframe defined in section 6.

Following are the important terms used in the model of data validation Solution:

* **Rule\_ID**- It represents the unique check definition id.
* **Rule\_Book\_ID** – It uniquely identifies the record in Rule\_book table.

The figure shown here shows the relationship between batch control model entities.



#### QAQC\_CATEGORY

This is the master table that defines various categories of checks that can be performed using QCF framework. Data stored in this table is static in nature and is manually populated before QCF ETL execution. E.g.: Attribute Data Match Check, Count Check, Duplicate Check etc.

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **PK** | **NULL?** | **DATA TYPE** |
| CATEGORY\_ID | YES | NO | BIGINT |
| CTGRY\_DESCR | NO | YES | VARCHAR |

#### QAQC\_RULES

The Rules table consists of all the checks that will be executed to measure a particular KPI. The rules have 2 set of SQL queries. In layman terms, the result of the 2 queries are compared with each other to validate the rule and identify number of passed and failed records. Data stored in this table is static in nature and is manually populated before QCF ETL execution.

| **COLUMN NAME** | **PK** | **NULL?** | **DATA TYPE** |
| --- | --- | --- | --- |
| RULE\_ID | YES | NO | BIGINT |
| DESCR | NO | YES | VARCHAR |
| TBL\_NAME | NO | YES | VARCHAR |
| TBL\_SCHM\_NAME | NO | YES | VARCHAR |
| SQL\_QRY1\_DB\_SRVR\_NAME | NO | YES | VARCHAR |
| SQL\_QRY1\_DB\_IN\_CTLG\_NAME | NO | YES | VARCHAR |
| SQL\_QRY1\_VAL | NO | YES | VARCHAR |
| SQL\_QRY2\_DB\_SRVR\_NAME | NO | YES | VARCHAR |
| SQL\_QRY2\_DB\_IN\_CTLG\_NAME | NO | YES | VARCHAR |
| SQL\_QRY2\_VAL | NO | YES | VARCHAR |

#### QAQC\_RULE\_BOOK

The Rule Book entity associates each of the defined rules to a particular QCF category & Function. It also consists information such as the Business Impact and the order in which the check needs to be executed for a particular job. Data stored in this table is static in nature and is manually populated before QCF ETL execution.

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **PK** | **NULL?** | **DATA TYPE** |
| RULE\_BOOK\_ID | YES | NO | BIGINT |
| FUNCTION\_ID | NO | NO | BIGINT |
| CATEGORY\_ID | NO | NO | BIGINT |
| RULE\_ID | NO | NO | BIGINT |
| EXECUTION\_ORDER | NO | NO | BIGINT |
| IS\_ACTIVE | NO | YES | BIT |
| BUSINESS\_IMPACT\_01\_DESCRIPTION | NO | YES | VARCHAR |
| BUSINESS\_IMPACT\_02\_DESCRIPTION | NO | YES | VARCHAR |
| IS\_PROCEDURE | NO | YES | BIT |

#### QAQC\_RULE\_CHECK\_RESULT

The Rule\_Check\_Result table stores the result of the check being performed for a particular batch\_id with a pass/fail indicator. These results or KPI are used for generating the technical validation report. Data is populated at runtime by QCF ETL package during technical validation.

| **COLUMN NAME** | **PK** | **NULL?** | **DATA TYPE** |
| --- | --- | --- | --- |
| RULE\_CHECK\_RESULT\_ID | YES | NO | BIGINT |
| BATCH\_ID | NO | YES | BIGINT |
| DQ\_RULE\_BOOK\_ID | NO | NO | BIGINT |
| CHECK\_SQL\_RESULT\_VALUE | NO | YES | NVARCHAR |
| COUNT\_VALUE | NO | YES | BIGINT |
| PERCENT\_OF\_TOTAL | NO | YES | SMALLINT |
| CHECK\_DATETIME | NO | NO | DATETIME |

#### QAQC\_FUNCTION

The Function entity stores information about different Functions for which check is being performed. Data stored in this table is static in nature and is manually populated before QCF ETL execution.

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **PK** | **NULL?** | **DATA TYPE** |
| FUNCTION\_ID | YES | NO | BIGINT |
| SBJCT\_AREA\_DESCR | NO | YES | VARCHAR |

#### 5.1.6. UPDATE\_DQ\_CHECK\_RESULT

This stored procedure accepts Batch\_ID as input parameter and updates the percentage of passed and failed checks in the QAQC\_Rule\_Check\_Result table. This stored procedure is dynamically executed by the QCF ETL package during technical validation.

* 1. Batch control Framework

While complete data load process from Source to Target, we will use the Batch Control framework along with QCF framework and Error handling framework to make sure that data is complete, consistent and accurate as expected. As part of batch control, we will persist the count of records processed at different layers during the data migration from landing area to CSV Extract load. Using set of specific audit tables populated through batch control framework, execution and monitoring of ETL jobs/packages will be handled.

#### 5.2.1. ETLLoadInfo

All SSIS packages have been designed to log execution information in audit tables through stored procedures. The ETLLoadInfo table holds details of each and every SSIS ETL package execution. This execution information can be uniquely identified using the batch\_id. Data is populated at runtime by the SSIS ETL package.

| **COLUMN NAME** | **PK** | **NULL?** | **DATA TYPE** |
| --- | --- | --- | --- |
| ETLLOADID | YES | NO | BIGINT |
| LOADID | NO | NO | BIGINT |
| ETL PHASE | NO | YES | NVARCHAR |
| PACKAGENAME | NO | YES | NVARCHAR |
| STATUS | NO | YES | NVARCHAR |
| PACKAGEID | NO | YES | NVARCHAR |
| PACKAGEVERSION | NO | YES | VARCHAR |
| STARTTIME | NO | YES | DATETIME |
| ENDTIME | NO | YES | DATETIME |
| EXECUTEDBY | NO | NO | NVARCHAR |

#### 5.2.2. ETLLoadSummary

The end to end load for every extract consists of Transform, Construct and QCF ETL Execution. This table captures all details of the end to end load and stores metadata of the end to end load. Data is populated at runtime by the SSIS ETL package.

| **COLUMN NAME** | **PK** | **NULL?** | **DATA TYPE** |
| --- | --- | --- | --- |
| LOADID | YES | NO | BIGINT |
| LOADNAME | NO | YES | NVARCHAR |
| ETL PHASE | NO | YES | NVARCHAR |
| STARTTIME | NO | YES | DATETIME |
| ENDTIME | NO | YES | DATETIME |
| LOADEXECUTIONSTATUS | NO | YES | NVARCHAR |
| TRANSFORMATIONBUILDSTATUS | NO | YES | NVARCHAR |
| TARGETTRANSFORMATIONCOMPLETIONDATE | NO | YES | DATETIME |
| ACTUALTRANSFORMATIONCOMPLETIONDATE | NO | YES | DATETIME |
| TARGETLOADDATE | NO | YES | DATETIME |
| ACTUALLOADDATE | NO | YES | DATETIME |
| LOADFILEGENERATED | NO | YES | BIT |
| FILELOADED | NO | YES | BIT |
| NUMBEROFRECORDSINLOADFILE | NO | YES | BIGINT |
| NUMBEROFRECORDSSUCCESSFULLYLOADED | NO | YES | BIGINT |
| NUMBEROFRECORDSFAILEDTOLOAD | NO | YES | BIGINT |
| TECHNICALVALIDATIONCOMPLETE | NO | YES | BIT |

### *ETL\_Execution\_Dependency*

The ETL\_Execution\_Dependency stores information about interdependencies between SSIS packages and also provides information about which ETL belongs to which ETL Phase. Data stored in this table is static in nature and is manually populated before data migration ETL executions.

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **PK** | **NULL?** | **DATA TYPE** |
| ETLEXECUTIONDEPENDENCYID | YES | NO | BIGINT |
| LOADNAME | NO | YES | NVARCHAR |
| PACKAGENAME | NO | YES | NVARCHAR |
| PARENTPACKAGENAME | NO | YES | NVARCHAR |
| CREATEDDATE | NO | YES | DATETIME |
| MODIFIEDDATE | NO | YES | DATETIME |
| ISACTIVE | NO | YES | BIT |
| ETL PHASE | NO | YES | NVARCHAR |

### *Row\_Count\_Info*

A row in this table provides information about no of records an SSIS package has processed. Data in this table is populated at runtime by the SSIS ETL package through stored procedure “INSERT\_ROW\_COUNT\_INFO”.

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **PK** | **NULL?** | **DATA TYPE** |
| ETLLOADID | NO | YES | BIGINT |
| INPUTROWCOUNT | NO | YES | BIGINT |
| INSERTROWCOUNT | NO | YES | BIGINT |
| UPDATEROWCOUNT | NO | YES | BIGINT |
| DELETEROWCOUNT | NO | YES | BIGINT |
| FAILEDROWCOUNT | NO | YES | BIGINT |

### 5.2.5. *INSERT\_AUDIT\_INFO Procedure*

This stored procedure accepts SSIS Package Name and ETL\_Phase as input parameters and generates a unique batch\_id for the ETL execution instance. The stored procedure populates data into ETLLoadInfo table and sets the execution status to “In Progress”.

### *5.2.6. UPDATE\_AUDIT\_INFO*

This stored procedure accepts Batch\_ID and ETL\_Execution\_Status as input parameters and updates the status of ETL execution instance in the ETLLoadInfo, ETLLoadSummary table. The stored procedure sets the execution status to “Complete” or “Failed”.

### 5.2.7*. INSERT\_ROW\_COUNT\_INFO*

This stored procedure row counts as input parameters and populates the count data processed along with error count for the ETL execution. The stored procedure populates data into Row\_Count\_Info table.

## 5.3. Exception handling framework

Exception handling plan highlights all actions and conventions necessary to handle exceptions that may arise during and after data migration cycle. Exceptions could be generated by SSIS ETL, Automated DQ Validation Or manual business reconciliation.

Technical exceptions will be stored in exception database and can be used for generating exception reports. Such technical exceptions can occur due to ETL errors or data inconsistencies and can be further categorised into exceptions generated by various ETL layers:

* Transform ETLs that transfer data from landing to Staging area
* Exceptions generated by Translate & Load tool (Hoover).

The technical validation as well as business validation errors would be accepted as a defect and then fixed by leveraging defect triage route.

Defect severity would be defined by stake holders during review. Refer section 3.2 for details.

### 5.3.1. AUDIT\_ERRORLOG

This table captures duplicate records, transformation, conversion, lookup & target data load errors detected during the transform ETL load. Data is populated at runtime by the SSIS ETL package.

|  |  |  |  |
| --- | --- | --- | --- |
| **COLUMN NAME** | **PK** | **NULL?** | **DATA TYPE** |
| ERROR\_ID | YES | NO | INT |
| BATCH\_ID | NO | YES | BIGINT |
| PACKAGE\_NAME | NO | YES | NVARCHAR |
| EXTRACT\_NAME | NO | YES | NVARCHAR |
| ALTERNATE\_KEY | NO | YES | NVARCHAR |
| ERROR\_COMPONENT\_NAME | NO | YES | NVARCHAR |
| ERROR\_COLUMN\_NAME | NO | YES | NVARCHAR |
| ERROR\_DESCRIPTION | NO | YES | NVARCHAR |
| ERROR\_TYPE | NO | YES | NVARCHAR |
| ERROR\_LOGGING\_TIME | NO | YES | DATETIME |

### 5.3.2. Identifying Exception

The following sections outline the exceptions that will be identified and how they can be handled

* Data inconsistency issue
  + data transformation errors - Update the mapping specification
  + type conversion issues - Technical Code fix / Update the mapping specification / data model changes
  + data integrity issues - Source data cleansing / Update the mapping specification
  + reference data lookup failure - Source data cleansing / Update the mapping specification
  + incorrect, incomplete or duplicated data – Source data cleansing
* DQ validation errors – Raise & triage the defects to identify a data fix (Source data cleansing / Technical Code fix)
  + Technical validation
  + Business reconciliation

### 5.3.3. Exception Classification and Resolution plan

Error severity would be defined by stake holders during review. Below are the severity level and corresponding criteria.

|  |  |
| --- | --- |
| Severity | Criteria |
| * Critical | * The error affects critical functionality, or a substantial portion of critical data is at a significant risk of loss or corruption. * The error leads to substantial loss of service or severely impacts business operations. * No work around available. |
| * Major | * Operations can continue in a restricted fashion, although long-term productivity might be adversely affected. * A major milestone is at risk. Ongoing and incremental data migration cycles are affected. * A temporary workaround is available. |
| * Minor | * The error impacts operations of some minor components but allows the user to continue using the downstream/dependent systems. * A milestone is at minimal risk. * Cosmetic issues, including errors in the documentation. |

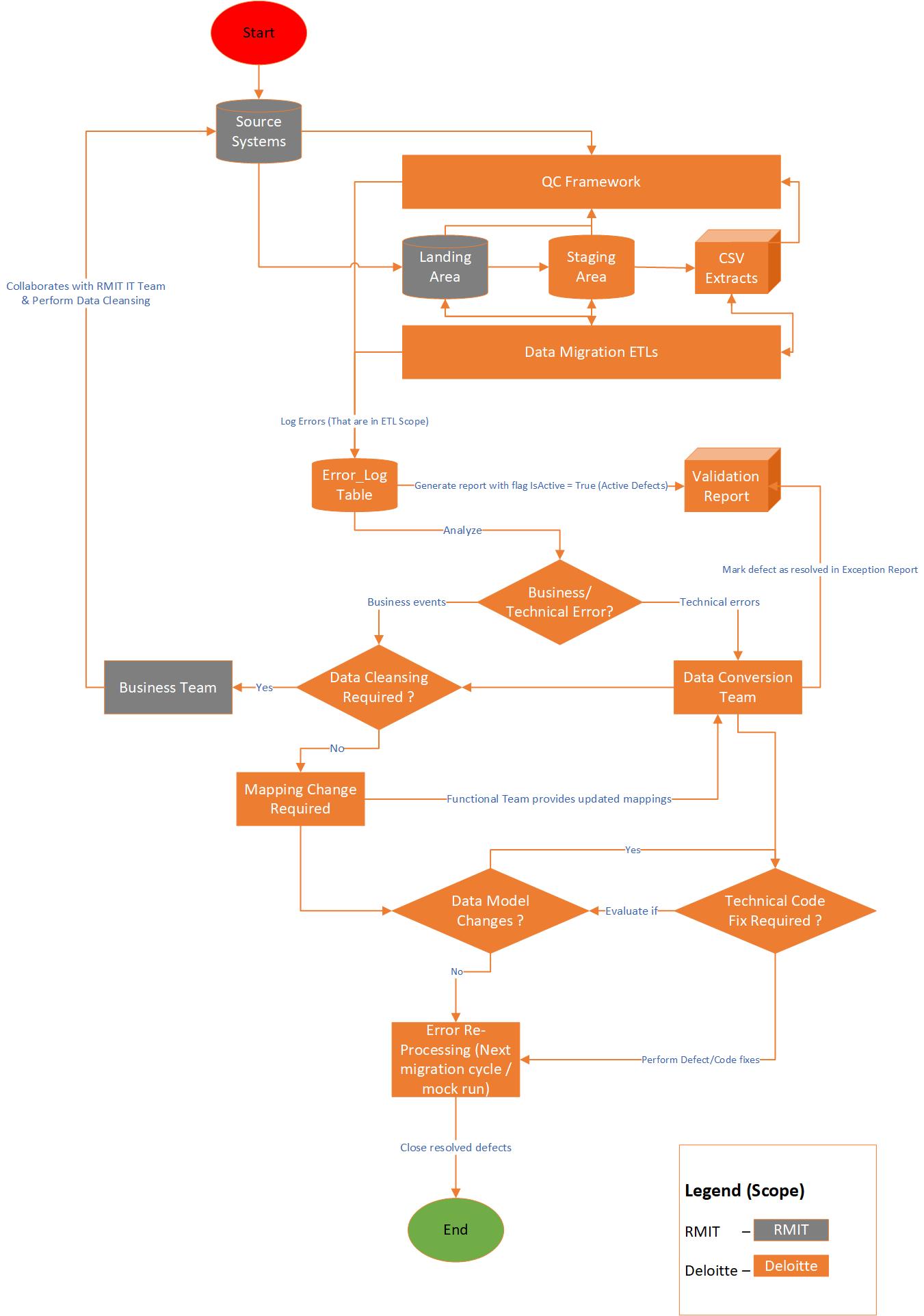
For each exception raised

* Stakeholders will need to be identified
* Discuss/Review exception fixing approach and
* Finalize action plan to clear the exception

| Sr No | Actions | Stake Holders | Resolution Approach |
| --- | --- | --- | --- |
| 1 | Source data cleansing | Deloitte | Data issues identified will be logged and shared with RMIT team. After cleaning up, relevant data will be loaded in the system once again |
| RMIT | RMIT team to clean up data at source end. |
| 2 | Update the mapping specification | Deloitte | Mapping issues//Change requests will be identified, logged, reviewed with RMIT and corrected. After fixing corresponding code issues relevant data will be loaded in the system once again |
| RMIT | Review and approve the changes/fixes |
| 3 | Technical code fix | Deloitte | After fixing technical code issues, relevant data will be loaded in the system once again |
| RMIT | Review and approve the data after the technical code fix |
| 4 | Data model changes | Deloitte | Data model issues/Change requests will be identified, reviewed with RMIT and corrected. After fixing corresponding mappings/code will be updated and relevant data will be loaded in the system once again |
| RMIT | Review and approve the changes/fixes |

### 5.3.4. Exception Correction and Reprocessing

The following diagram illustrates steps that would be taken to identify and resolve exceptions/data errors. After an assessment of errors, corrections will need to be made in the source and the corrected data will be reprocessed during next Migration cycle. Once the error records have been fixed at the source, the error records will need to be manually deleted from the error tables, before execution of next migration cycle.



1. Technical Validation Reports

The technical validation reports will be put in place to showcase the effectiveness of the data conversion solution

* Count Validation and Error Record counts:
  + There are two types of count validation reports 1st will showcase Landing record counts (transformed and restricted data based on business rules) vs Staging record count for each extract and 2nd type of reports will showcase Staging area count vs Workday reports count. Any mismatch will have to be analysed manually and corrected if required
  + Other errors are also logged as part of Batch control framework. Some common examples for these errors can be:
    - Source data length is greater than defined target attribute resulting in data truncation Or
    - Referential Integrity failure indicating lookup failures. Issues will have to be analysed manually and corrected if required

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Subject Area | Extract Name | Input Row Count | Target Row Count | % loaded successfully | Error Record Count (Eliminated data) | | Error record % | Error Record Count ( No data elimination) |
|  | Stage table name | Landing table count after application of all filter criteria | Final Target Row count in staging tables | Records loaded successfully in Staging area | Error records that resulted in data elimination | | Error percentage | Error identified during QC validation |
| HR-AU | APPLICANTS | 500 | 200 | 40% | | 300 | 60% | 100 |
| HR-AU | CREATE POSITION | 1000 | 1000 | 100% | | 100 | 10% | 0 |
| HR-AU | HIRE EMPLOYEES | 100 | 100 | 100% | | 0 | 0% | 0 |

Error records while loading from landing to stage and from stage to CSV files can be captured in a detailed report for verification

| Subject Area | Extract Name | Alternate Key Columns | Alternate Key Value | Stage Column Name | Error description | Error Type |
| --- | --- | --- | --- | --- | --- | --- |
|  | Stage table name | Field name used to uniquely identify a row | Unique record identifier | Field name that has error |  | Identify if the error is from transform stage or from conform stage |
| HR-VN | HIRE EMPLOYEES | WORKER\_ID | v91201 | END\_EMPLOYMENT\_DATE | The data value cannot be converted for reasons other than sign mismatch or data overflow. | Destination Error |
| HR-AU | EMPLOYEE PERFORMANCE REVIEWS | EMPLOYEE\_ID | E00000 | MANAGER\_ID | A truncation caused a row to be redirected, based on the truncation disposition settings. | Conformance Error |
| HR-AU | EMPLOYEE PERFORMANCE REVIEWS | EMPLOYEE\_ID | HRADMIN | EMPLOYEE\_ID | NULL or Blank value present in required field | REQUIRED\_FIELDS |
| HR-VN | HIRE EMPLOYEES | WORKER\_ID | v91201 | END\_EMPLOYMENT\_DATE | NULL or Blank value present in required field | REQUIRED\_FIELDS |

* Duplicate checks are also captured as part of Exception framework while making sure that the target data is clean. These records can be traced back to source using the information available in Alternate key column available in the reports. If multiple columns from an extract’s alternate key , they are merger together in error framework with a pipe delimiter in between for logical segregation.

|  |  |  |  |
| --- | --- | --- | --- |
| Subject Area | Extract Name | Alternate Key Columns | Alternate Key |
|  | Stage table name | Field name used to uniquely identify a row | Unique record identifier |
| HR-AU | REQUEST COMPENSATION CHANGE | EMPLOYEE\_ID ,ALLOWANCE\_PLAN\_ID | vXXXXX |
| HR-VN | CHANGE LEGAL NAME | WORKER\_ID | vXXXXX |
| HR-VN | MAINTAIN CONTACT INFORMATION ADDRESS | WORKER\_ID, ADDRESS1, ADDRESS2 | vxxxxx~|NXX xx 1, xxxx 1F, xxxxxxxxx xxxxxx |

Data validation framework also captures the following information

* Transformation logic validation as per mapping specification

| Subject Area | Extract Name | Alternate Key Columns | Alternate Key Value | Stage Column Name | Error description | Error Type |
| --- | --- | --- | --- | --- | --- | --- |
|  | Stage table name | Field name used to uniquely identify a row | Unique record identifier | Field name that has error |  | Identify if the error is from transform stage or from conform stage |
| HR-VN | HIRE EMPLOYEES | WORKER\_ID | v91201 | END\_EMPLOYMENT\_DATE | The data value cannot be converted for reasons other than sign mismatch or data overflow. | Destination Error |

* Required column check

| Subject Area | Extract Name | Alternate Key Columns | Alternate Key Value | Stage Column Name | Error description | Error Type |
| --- | --- | --- | --- | --- | --- | --- |
|  | Stage table name | Field name used to uniquely identify a row | Unique record identifier | Field name that has error |  | Identify if the error is from transform stage or from conform stage |
| HR-VN | HIRE EMPLOYEES | WORKER\_ID | v91201 | END\_EMPLOYMENT\_DATE | NULL or Blank value present in required field | REQUIRED\_FIELDS |

1. Appendix

## References

|  |  |  |
| --- | --- | --- |
| **No.** | **Name** | **File** |
| 1 | High-Level Data Conversion Plan | [**Data Conversion Plan**](https://rmiteduau.sharepoint.com/:w:/r/sites/erprepo/Technology%20%20Phase%202/2.%20Data/Plan%20and%20Architect/CNV%20020%20Data%20Conversion%20Plan/Pi_DTA_Data%20Conversion%20Plan.docx?d=w7fe4e8bb23ef471890af46c76be26dc8&csf=1&e=QYnTky) |
| 2 | Pi\_DTA\_Data\_Conversion\_Strategy |  |
|  |  |  |

## QCF Entity column definitions

The following table lists each of the entities/attributes and provides a brief definition and a sample list of value



### DQ.CATEGORY

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | COLUMN DEFINITION | SAMPLE VALUES |
| CATEGORY\_ID | UNIQUELY IDENTIFIES THE CATEGORY ID | 1,2,3 |
| CTGRY\_DESCR | DESCRIBES THE CATEGORY | Attribute\_Data\_Match, Count\_Check, Duplicate\_Check |

### DQ.RULES

| COLUMN NAME | COLUMN DEFINITION | SAMPLE VALUES |
| --- | --- | --- |
| RULE\_ID | UNIQUELY IDENTIFIES THE CHECK | 101,102,103 |
| DESCR | DESCRIBES THE CHECK | DC\_Applicants , CC\_Applicants |
| TBL\_NAME | REPRESENT THE TABLE NAME ON WHICH CHECK IS BEING PERFORMED | DIMCUSTOMER, DIMACCOUNT |
| TBL\_SCHM\_NAME | REPRESENT THE SCHEMA NAME ON WHICH CHECK IS BEING PERFORMED | DBO |
| SQL\_QRY1\_DB\_SRVR\_NAME | REPRESENT THE SERVER NAME WHERE QUERY 1 IS SUPPOSED TO BE EXECUTED | SRVR001, SRVR002 |
| SQL\_QRY1\_DB\_IN\_CTLG\_NAME | REPRESENT THE DATABASE NAME WHERE QUERY 1 IS SUPPOSED TO BE EXECUTED | ADVENTUREWORKSDB |
| SQL\_QRY1\_VAL | REPRESENT THE QUERY RELATED TO THE QUERY BEING PERFORMED | SELECT COUNT(\*) FROM TABLE\_NAME |
| SQL\_QRY2\_DB\_SRVR\_NAME | REPRESENT THE SERVER NAME WHERE QUERY 2 IS SUPPOSED TO BE EXECUTED | SRVR001, SRVR002 |
| SQL\_QRY2\_DB\_IN\_CTLG\_NAME | REPRESENT THE DATABASE NAME WHERE QUERY 2 IS SUPPOSED TO BE EXECUTED | ADVENTUREWORKSDB |
| SQL\_QRY2\_VAL | REPRESENT THE QUERY RELATED TO THE QUERY 2 BEING PERFORMED | SELECT COUNT(\*) FROM TABLE\_NAME |

### DQ.RULE\_BOOK

| COLUMN NAME | COLUMN DEFINITION | SAMPLE VALUES |
| --- | --- | --- |
| RULE\_BOOK\_ID | UNIQUELY IDENTIFIES THE RULE BOOK ITEMS | 1000, 1001,1002 |
| FUNCTION\_ID | REPRESENT THE SUBJECT AREA ID IN DQ.FUNCTION TABLE | 1000 |
| CATEGORY\_ID | REPRESENT THE CATEGORY ID IN DQ.CATEGORY TABLE | 1003 |
| RULE\_ID | REPRESENT THE CHECK DEFINITION ID IN DQ.RULES TABLE | 1001 |
| EXECUTION\_ORDER | SPECIFIES THE ORDER IN WHICH CHECK NEEDS TO BE EXECUTED FOR A PARTICULAR JOB | 1,2,3 |
| IS\_ACTIVE | SPECIFIES IF THE CHECK IS ACTIVE OR INACTIVE. ‘N’ MEANS INACTIVE AND ‘Y’ MEANS ACTIVE | 1,0 |
| BUSINESS\_IMPACT\_01\_DESCRIPTION | DESCRIBES THE BUSINESS IMPACT OF THE CHECK PERFORMED | THE SALES ORDER RECORD IS NOT ACCURATE. WE ARE MISSING OR EXCEEDING THE NUMBER OF RECORDS. |
| BUSINESS\_IMPACT\_02\_DESCRIPTION | DESCRIBES THE BUSINESS IMPACT OF THE CHECK PERFORMED | THE SALES ORDER RECORD IS NOT ACCURATE. WE ARE MISSING OR EXCEEDING THE NUMBER OF RECORDS. |
| IS\_PROCEDURE | SPECIFIES IF THE QUERY IS A STORED PROCEDURE | 1,0 |

### DQ. RULE\_CHECK\_RESULT

| COLUMN NAME | COLUMN DEFINITION | SAMPLE VALUES |
| --- | --- | --- |
| RULE\_CHECK\_RESULT\_ID | UNIQUELY IDENTIFIES THE DQ CHECK RESULTS | 1000, 1001,1002 |
| ETLLOADID | REPRESENTS THE ETLLOADID FROM AUDIT.ETLLOADINFO TABLE | 1000, 1001,1002 |
| DQ\_RULE\_BOOK\_ID | STORES THE UNIQUE RULE BOOK RECORD | 1001, 1002 |
| CHECK\_SQL\_RESULT\_VALUE | SPECIFIES WHETHER THE CHECK IS PASS OR FAIL. VALUE ‘F’ MEANS CHECK IS FAILED AND ‘P’ MEANS PASSED | 1,0 |
| COUNT\_VALUE | REPRESENTS NO OF RECORDS CORRESPONDING TO EITHER FAILED OR PASSED RESULT | 3000, 100 |
| PERCENT\_OF\_TOTAL | REPRESENTS A PERCENTAGE VALUE RANGING FROM 0-100 INDICATING HOW MUCH RECORDS SUCCESSFULLY PASSED OR FAILED VALIDATION | 100% |
| CHECK\_DATETIME | REPRESENTS DATE TIME WHEN THE CHECK WAS PERFORMED | 10-JULY-2019 17:00:00 |

### DQ.FUNCTION

|  |  |  |
| --- | --- | --- |
| COLUMN NAME | COLUMN DEFINITION | SAMPLE VALUES |
| FUNCTION\_ID | UNIQUELY IDENTIFIES THE CHECK | 101,102,103 |
| DESCR | DESCRIBES THE SUBJECT AREA | HR, FINANCE |