

**Merchandise Planning**

**Build**

**Technical Solution Framework**

**Enterprise Planning (EP)**

Version: 0.14

Issue Date: 27/11/2013

Author: Tushar Pradhan/Amit Nikam/Piyush Pradhan

Quality Plan:

|  |  |
| --- | --- |
| Quality Role | Programme Role / Name |
| Document Owner | Amit Nikam |
| Author(s) | Piyush Pradhan (JDA),  Tushar Pradhan & RajendraPrasad Yadav &Amit Nikam(Capgemini) |
| Reviewer(s) | Ronald Kohn(Capgemini) |
| Approver(s) MNS | Gordon Cox |
| Approver(s) JDA | Piyush Pradhan(JDA) |

Document History:

The following versions of the document have been produced:

| Version | Status (Draft / Final) | Issue Date | Author | Reason for Issue |
| --- | --- | --- | --- | --- |
| 0.1 | Draft | 12/08/2013 | Tushar Pradhan & RajendraPrasad Yadav | Initial draft |
| 0.1 | Draft | 13/08/2013 | Piyush Pradhan | Updates for JDA. Section 2 to Section 4.1 |
| 0.1 | Draft | 21/08/2013 | Tushar Pradhan | Updated as per comments received from Denis Mooney |
| 0.2 | Draft | 02/10/2013 | Amit Nikam | Updated section 1 to 5. |
| 0.3 | Draft | 23/10/2013 | Amit Nikam | Updated the interface tables, Error tables as per Ron’s Comments |
| 0.4 | Draft | 24/10/2013 | Amit Nikam | Updated document after discussion with Ron. |
| 0.5 | Draft | 25/10/2013 | Amit Nikam | Updated after discussion with Piyush and Ron |
| 0.6 | Draft | 15/11/2013 | Amit Nikam | Updated after discussion with Jin, John Courtney and Sandeep Kumar. |
| 0.7 | Draft | 21/11/2013 | Amit Nikam | Updated document for automation scripts after discussion with Steven Murray which will be developed by JDA and some comments from Piyush. Also added the log details for server process. |
| 0.8 | Draft | 22/11/2013 | Amit Nikam | Added the Fact transactional interface table. |
| 0.9 | Draft | 27/11/2013 | Amit Nikam | Updated document after review comments from Ron and added (DILO and E0259) link. |
| 0.10 | Draft | 18/12/2013 | Amit Nikam | Updated document based on review comments from Lee and Dennis |
| 0.11 | Draft | 10/02/2014 | Amit Nikam and Sandeep Kumar | Updated document based on review comments from Gordon and Piyush. |
| 0.12 | Draft | 20/02/2014 | Sandeep Kumar | Updated the data integrity section further after discussing with Ron. |
| 0.13 | Draft | 25/02/2014 | Amit Nikam | Updated sections after comments from Gordon. |
| 0.14 | Draft | 03/03/2014 | Amit Nikam | Updated Document after Gordon comments on trans\_id |

Open Questions:

| # | Question/Open Design Topic | Issue Date | Responsible | Resolution |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Document Reference:

| # | Document Name | Sharepoint Location |
| --- | --- | --- |
| 1 | Batch Schedule | [https://mnscorp.sharepoint.com/sites/GM4ProgrammeTeamsite/02\_MP/Forms/AllItems.aspx?RootFolder=%2Fsites%2FGM4ProgrammeTeamsite%2F02\_MP%2F04%20Build%2F01%20Deliverables%2FMPB01%20-%20Batch%20Schedule%2F03%20Issued%20For%20Approval&FolderCTID=0x0120006CFDFB756210D740899B133F8C129BBB&View={3F451D94-EE87-4C65-B753-5E83FB6E03F4}&InitialTabId=Ribbon%2EDocument&VisibilityContext=WSSTabPersistence](https://mnscorp.sharepoint.com/sites/GM4ProgrammeTeamsite/02_MP/Forms/AllItems.aspx?RootFolder=%2Fsites%2FGM4ProgrammeTeamsite%2F02_MP%2F04%20Build%2F01%20Deliverables%2FMPB01%20-%20Batch%20Schedule%2F03%20Issued%20For%20Approval&FolderCTID=0x0120006CFDFB756210D740899B133F8C129BBB&View=%7b3F451D94-EE87-4C65-B753-5E83FB6E03F4%7d&InitialTabId=Ribbon%2EDocument&VisibilityContext=WSSTabPersistence) |
| 2 | Technical Configuration Document | [https://mnscorp.sharepoint.com/sites/GM4ProgrammeTeamsite/02\_MP/Forms/AllItems.aspx?RootFolder=%2Fsites%2FGM4ProgrammeTeamsite%2F02\_MP%2F03%20Detailed%20Design%2F01%20Deliverables%2FMPDD42%20-%20MP%20Technical%20Configuration%20Design%20Document%2F03%20Issued%20For%20Approval&FolderCTID=0x0120006CFDFB756210D740899B133F8C129BBB&View={3F451D94-EE87-4C65-B753-5E83FB6E03F4}&InitialTabId=Ribbon%2EDocument&VisibilityContext=WSSTabPersistence](https://mnscorp.sharepoint.com/sites/GM4ProgrammeTeamsite/02_MP/Forms/AllItems.aspx?RootFolder=%2Fsites%2FGM4ProgrammeTeamsite%2F02_MP%2F03%20Detailed%20Design%2F01%20Deliverables%2FMPDD42%20-%20MP%20Technical%20Configuration%20Design%20Document%2F03%20Issued%20For%20Approval&FolderCTID=0x0120006CFDFB756210D740899B133F8C129BBB&View=%7b3F451D94-EE87-4C65-B753-5E83FB6E03F4%7d&InitialTabId=Ribbon%2EDocument&VisibilityContext=WSSTabPersistence) |
|  |  |  |
|  |  |  |

Glossary

| # | Term | Definition |
| --- | --- | --- |
| 1 | EP | Enterprise Planning |
| 2 | RICEF | Reports, interface, conversions, enhancements and forms |
| 3 | EKB | Enterprise Knowledge Base |
| 4 | ERD | Environment Requirements Definition document |
| 5 | EDW | Enterprise Data Warehouse |
| 6 | SAP ECC | Enterprise Central Component – this is where all the M&S transactions are captured |
| 7 | SAP BI | Business Intelligence |
| 8 | SCOM | System Center Operations Manager |
| 9 | Netcool | System that collects and consolidates enterprise-wide event and alarm information in real-time from many different network data sources and presents a simplified view of this information to operators and administrators |

**Contents**

[1 Introduction 7](#_Toc381784288)

[1.1 Document Purpose 7](#_Toc381784289)

[1.2 Document Scope 8](#_Toc381784290)

[1.3 Intended Audience 9](#_Toc381784291)

[2 Application Context 10](#_Toc381784292)

[3 Logical Design 11](#_Toc381784293)

[3.1 Overview 11](#_Toc381784294)

[3.2 Interface Tables 13](#_Toc381784295)

[3.2.1 Structure Data 15](#_Toc381784296)

[3.2.1.1 Data Integrity 16](#_Toc381784297)

[3.2.1.2 Data Resilience 17](#_Toc381784298)

[3.2.2 Transaction Data 17](#_Toc381784299)

[3.2.2.1 Data Integrity 18](#_Toc381784300)

[3.2.2.2 Data Resilience 19](#_Toc381784301)

[3.2.2.3 Transaction ID logic for BAU data load vs Conversions 19](#_Toc381784302)

[3.3 Framework Components 20](#_Toc381784303)

[3.4 Core Application Components 21](#_Toc381784304)

[3.4.1 EP Tables 21](#_Toc381784305)

[3.4.2 Processes 21](#_Toc381784306)

[3.4.3 Services 21](#_Toc381784307)

[4 Physical Design 23](#_Toc381784308)

[4.1 Overview 23](#_Toc381784309)

[4.2 File Systems & Schemas 24](#_Toc381784310)

[4.2.1 Subversion Folder Structure 24](#_Toc381784311)

[4.2.2 Windows File System 24](#_Toc381784312)

[4.2.3 Oracle Schema 25](#_Toc381784313)

[4.2.4 Batch operations 26](#_Toc381784314)

[4.2.4.1 Standard Return Codes 26](#_Toc381784315)

[4.2.4.2 Standard file/table operation codes 26](#_Toc381784316)

[4.2.4.3 Configuration files 26](#_Toc381784317)

[4.3 Process Scheduler (Windows) 26](#_Toc381784318)

[4.4 Auditing and Logging 28](#_Toc381784319)

[4.4.1 Distributed Logs 28](#_Toc381784320)

[4.4.1.1 Audit logs 28](#_Toc381784321)

[4.4.1.2 Detailed Logs 31](#_Toc381784322)

[4.4.2 Oracle Logs 33](#_Toc381784323)

[4.5 Data Retention\Replay 33](#_Toc381784324)

[4.5.1 Data Retention Approach 33](#_Toc381784325)

[4.5.2 Data Replay Approach 34](#_Toc381784326)

[4.6 Error Handler 34](#_Toc381784327)

[4.7 Report Generator 35](#_Toc381784328)

[4.7.1 User Notifications 35](#_Toc381784329)

[4.7.2 Exception Reports (Warnings) 35](#_Toc381784330)

[4.8 Housekeeper 35](#_Toc381784331)

[4.9 Reusable Routines 36](#_Toc381784332)

[5 Back-Ups 37](#_Toc381784333)

[5.1 Oracle Backups 37](#_Toc381784334)

[5.2 Windows Backups 37](#_Toc381784335)

# Introduction

## Document Purpose

The purpose of this document is to specify the software/technical framework that will support the Reports, Interfaces, Conversions, Enhancements and Forms (RICEF) Object components required by the Marks and Spencer (M&S) implementation of the JDA Enterprise Planning module.

This document gives information about:

1. Overview of JDA Enterprise Planning (EP) Architecture.
2. Components involved in JDA EP.
3. JDA Database server Process and JDA EP client automation Batch Process involved in JDA EP.
4. Framework, Error and Exception Handling (including logging, alerting and replay), data retention, archiving/housekeeping and scheduling involved in overall development of RICEF.

## Document Scope

The scope of this document is to define the software/technical solution framework for Enterprise Planning JDA EP/EKB and will not detail out any integration of the source system and any changes which have been made outside JDA EP/EKB.

The MP Solution Framework describes how data will be processed in and out of the solution. There are Core Application Components such as JDA load utilities that are executed by a controlled batched schedule. For these JDA utilities to be scheduled and batched, it is necessary to package or wrap them. By wrapping them it is possible to enable sequencing, calling, tracing, and logging. It also provides the ability to perform housekeeping and replay.

The MP Solution Framework adheres to the details contained in both Functional and Technical Configuration documents. The processing of both transaction and master data from staging tables to live tables is described in the MP Solution Framework.

Application management will leverage this document to develop documentation for support.

This document assumes the deployment of a single batch server based on the specification detailed in section [4.2](#_File_Systems_&). Although the application deployment is expected to include multiple mid-tier application servers, batch deployment assumes a single instance/server.

## Intended Audience

The intended audience for this document are the developers, data conversion team, integration team, testing team and batch scheduling team.

# Application Context

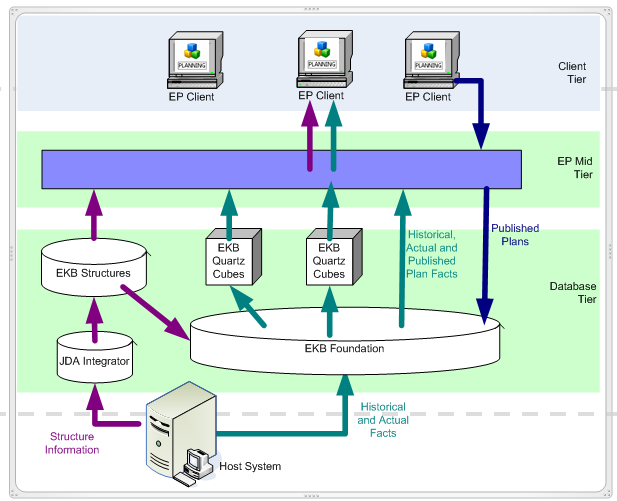
The following is an application context only. This context is meant to convey integration and boundaries between all layers. This context is not an architectural diagram and should not be used for application specific deployment. Application details are covered in the appropriate Environment Requirements Definition document (ERD).



Figure 1 Application Context

# Logical Design

## Overview



External

Systems

Figure 2 EP Architecture

A simplified view of JDA Enterprise Planning is described below:

JDA Enterprise Planning is a Client Server architecture set up. Figure 2. shows the different components of the EP architecture. JDA EP consists of Database, Application and the Components consist of:

* EP Client
* Enterprise Knowledge Base (EKB)
* Central repository of structure (EKB)
* Central repository of fact data (EKB)
* Enterprise Planning Mid-Tier (EP Mid-Tier)
* JDA Integrator
* Cubes (Quartz)
* EP Server Administrator Workbench
* Host System represents those systems acting as source data and/or system of record (e.g. SAP ECC, SAP BI, EDW and their corresponding integration layers)

The Mid-Tier application handles the request between the EP Client and EKB. The Mid-Tier can be configured to have multiple Application machines set up in order to handle the load based on the number of EP Clients set up on the system.

Cubes (Quartz) are built to handle data aggregation and storage associated for a particular group of plans. This makes the process to deal with large set of data and generate results in expected time.



Figure 3 EP Components for RICEF

Figure 3. Shows the different components involved for development of EP RICEF objects which are described below.

## Interface Tables

Master data sent by source systems are written into a separate schema. This separate schema is controlled by JDA Integrator (JI). This integration layer leverages 3rd party components from IBM DataStage. JDA Processing includes either JI utilities or JDA proprietary components.

Transaction data sent by source systems are always written to JDA Staging Tables. The Staging Tables are a mirror of the JDA Live Tables (FACT Tables) excluding the Transaction ID and Sequence Number columns.

The staging tables are prefixed with “PKB\_”. The update type of each of these tables is described in the Technical Configuration Document.

Wrappers will leverage JI utilities to process master data and JDA components to process transaction data and master data.

Interface tables will be held in two schemas of the EKB database

The schemas are:

* EXT:

Interface tables which will hold the Structure data will be held in this schema

* EKB:

Interface tables which will hold the Fact transactional data will be held in this schema

### Structure Data

The interface tables for Structure data can be classified in 4 groups based on the type of data they contain. These are:

1. **Product**

The tables contained in this group are used to interface all the product hierarchy data. Below are the tables that will be created for holding the product hierarchy data.

* AG\_1\_ARTICLE
* AG\_1\_RANGE
* AG\_1\_SUB\_DEPARTMENT
* AG\_1\_T\_DEPARTMENT
* AG\_1\_CATEGORY
* AG\_1\_BUSINESS\_UNIT
* AG\_1\_TOTAL\_GM

1. **Organization (Location)**

The tables contained in the group are used to interface all the organization hierarchy data. Below are the tables that will be created for holding the organization hierarchy data.

* AG\_2\_TOUCHPOINT
* AG\_2\_CHANNEL
* AG\_2\_COUNTRY
* AG\_2\_COUNTRY\_GROUP
* AG\_2\_OPERATING\_REGION
* AG\_2\_ZONE
* AG\_2\_OPERATING\_MODEL
* AG\_2\_TOTAL
* AG\_2\_ECOM\_GROUP
* AG\_2\_TOTAL\_ECOM
* AG\_2\_REGION
* AG\_2\_DIVISION
* AG\_2\_COUNTRY\_
* AG\_2\_SHAPE\_OF\_CHAIN
* AG\_2\_PLANNING\_LEVEL
* AG\_2\_COUNTRY\_LOCATION
* AG\_2\_CHANNEL\_LOCATION

1. **Calendar**

The tables contained in the group are used to interface all the calendar hierarchy data. Below are the tables that will be created for holding the calendar hierarchy data.

* AG\_4\_DAY
* AG\_4\_WEEK
* AG\_4\_MONTH
* AG\_4\_QUARTER
* AG\_4\_SUPER\_SEASON
* AG\_4\_YEAR
* AG\_4\_FINANCIAL\_HALF
* AG\_4\_FINANCIAL\_YEAR
* AG\_4\_PHASE
* AG\_4\_SEASON
* AG\_4\_SUPERSEASON

1. **Relation Table**

The below table is used to interface the relationship between the child and parent members in the hierarchy in the same business entity (i.e. Products, Calendar and Organization).

* EXT\_EPSD\_IMPORT\_MEMBER\_REL

Primary Key (For all the above tables) – INPUTSEQUENCE

The sending system should respect the primary key for structure data if the table is not empty. The record will fail insertion if the input sequence is repeated for multiple records. If the same record is inserted with two different input sequences then the records will be processed sequentially.

Errors occurring while moving data from the staging/interface tables to the live tables will be held in the error table for structure data i.e. . Section [4.4.1](#_Windows_Logs) details the names of the error tables where the errors are stored.

#### Data Integrity

Structure data records rejected by JDA will be recorded and logged in the error tables. These can be used to validate integrity between the sending system and receiving application (JDA).

If records are not rejected by JDA then it is assumed that there is no discrepancy between the sending system and receiving application (JDA).

DataHub (Datastage) 🡪 JDA Staging tables (EXT Schema): If records are rejected/fail insertion into the staging table then datastage receives an error message and continues with the process as specified in the functional specification for the interface. The default mechanism is to continue processing the subsequent records if records are rejected.

JDA Staging tables (EXT schema) 🡪 JDA Live tables (EKB Schema): If records are rejected by the JDA live tables then the JDA Integrator (JI) jobs write the rejected records into the error table - EXT\_EPSD\_ERROR.

There is the ability to determine the number of records not written to the destination tables. This ability exists for both the legs i.e. from Datahub/Datastage 🡪 Staging tables and the processing from the JDA staging tables to the JDA Live tables.

#### Data Resilience

The sending application is responsible for sending an ‘action code’ to determine the type of action to be performed on the record i.e. add, update or delete. If the sending system sends an ‘Update- type 2’ record then the record is updated only if a record originally existed in the live table. The ‘type 2’ record will fail if the record is not already present in the live tables.

### Transaction Data

The Fact tables hold transaction data and can be grouped into a number of groups based on the kind of data they contain. Below are the interface tables for Fact Transactional data

1. **Sales**

* PKB\_ACTUAL\_SALES\_1

1. **Intake**

* PKB\_ACTUAL\_INTAKE

1. **Inventory**

* PKB\_ACTUAL\_DC\_INV
* PKB\_ACTUAL\_STR\_INV

1. **Adjustments and Markdowns**

* PKB\_ACTUAL\_DC\_ADJ
* PKB\_ACTUAL\_STR\_ADJ
* PKB\_ACTUAL\_DC\_MKDN
* PKB\_ACTUAL\_STR\_MKDN
* PKB\_ACTUAL\_DC\_SIS
* PKB\_ACTUAL\_STR\_SIS

1. **Outlet Stock Transfer**

* PKB\_ACTUAL\_OL\_TFR

1. **Return to Warehouse**

* PKB\_ACTUAL\_RTN

1. **Stock Allocation**

* PKB\_ACTUAL\_ALLOC

1. **Lift and Shift**

* PKB\_ACTUAL\_LIFT\_SHIFT

1. **VAT**

* PKB\_ACTUAL\_VAT

1. **Like for Like Flag**

* PKB\_ACTUAL\_FLAG

1. **UBOC**

* PKB\_ACTUAL\_UBOC

1. **Shipment**

* PKB\_ACTUAL\_SHPMNT

All the transactional tables have a unique index built on the following fields:

* TRANS\_ID
* SEQ
* PRODUCT\_NAME
* ORGANIZATION\_NAME
* CALENDAR\_NAME

The combination of these fields should always be unique. The sending system should respect this unique combination for transactional data if the table is not empty. Duplicate records will be rejected at the table level and the sending system/tool will receive an error message.

Errors occurred while moving data from the staging table to the live tables will be held in the error tables specific to the fact tables. Section [4.4.1](#_Windows_Logs) details the names of the error tables where the errors are stored.

#### Data Integrity

Transactional data records rejected by JDA will be recorded and logged in the error tables. These can be used to validate integrity between the sending system and receiving application (JDA).

If records are not rejected by JDA then it is assumed that there is no discrepancy between the sending system and receiving application (JDA).

Datastage 🡪 JDA Staging tables (EKB Schema): If records are rejected/fail insertion into the staging table then datastage receives an error message and continues with the process as specified in the functional specification for the interface. The default mechanism is to continue processing the subsequent records if records are rejected.

JDA Staging tables (EKB schema) 🡪 JDA Live tables (EKB Schema): If records are rejected by the JDA live tables then the fact update job writes the rejected records into the respective FTF error tables.

There is the ability to determine the number of records not written to the destination tables. This ability exists for both the legs i.e. from Datastage 🡪 Staging tables and the processing from the JDA staging tables to the JDA Live tables.

#### Data Resilience

For transactional data there is not transaction type/action code equivalent. Therefore it is assumed that updates to previously loaded data to the same unique constraints for product (article), organisation (site) and calendar (week ending date) contain the updated transaction value. If a record is present already in the live table then it will be updated and it will be created in the live table if not present already.

#### Transaction ID logic for BAU data load vs Conversions

The TRANS\_ID could be used to identify if the staging records created during migration were BAU or Conversion. It will be a concatenation of the date and batch number whereby the range from 90 to 99 is reserved for BAU:

* A date of February 6th, 2014 would lead to ‘140206’ + batch number (00, 01 etc.) so 14020600

During conversions the TRANS\_ID is filled as follows:

* Fill transaction ID with the load date + batch number. A date of February 6th, 2014 would lead to ‘140206’ + batch number (00, ,01,02 etc.)
* In case large numbers of records are to be loaded, the conversion software needs to create multiple batches, so 140202600, 14020601 etc.
* As batch numbers 90 to 99 are reserved for BAU, the maximum number of batches per conversion date is 90 (batch number 00 up to 89)
* Conversions will not run on Sundays when BAU is also running.

BAU interfaces fill the TRANS\_ID as follows:

* The batch number will always be a nine so that a TRANS\_ID can be quickly recognized as a BAU load. So the transaction ID for coming Sunday would be 14020990 (February 9th, 2014 plus a ‘90’)

The following example shows how staging records from 2 Conversion batches (batch 0 and 1) look like:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Staging Table | | | | | |
| TRANS\_ID | SEQ | PRODUCT\_NAME | ORGANIZATION\_NAME | CALENDAR\_NAME | Closing Stock |
| 14020600 | 1 | A2020221 | TP\_1234 | W20130122 | NULL |
| 14020601 | 1 | A2020221 | TP\_1234 | W20130129 | 40 |

Doing the above will also facilitate with any corrections and replay required to be done for a particular data set while it is still in the JDA Staging Tables and the data hasn’t been processed to the JDA Live tables.

Corrections: If we have an incorrect load for a particular batch and the associated staging records have not been processed yet, we can identify the records in the Staging table with the TRANS\_ID and delete them using a simple SQL statement.

Replay: If staging records have already been processed and corrections are required, the sending system will send a new batch with the metrics for the identified combination/s (touchpoint / article / week). The new staging records will get a new TRANS\_ID as per the logic stated above.

It is assumed that any master data or transactional data in the appropriate EXT and EKB staging table will be processed during normal BAU Batch. Because the BAU batch cannot distinguish conversion data and because the staging tables are shared, it is assumed that processing of non BAU data will happen outside of the normal weekend batch times

The following link will contain the Technical Configuration document for EP which will detail the definition of the interface tables

[https://mnscorp.sharepoint.com/sites/GM4ProgrammeTeamsite/02\_MP/Forms/AllItems.aspx?RootFolder=%2Fsites%2FGM4ProgrammeTeamsite%2F02\_MP%2F03%20Detailed%20Design%2F01%20Deliverables%2FMPDD42%20-%20MP%20Technical%20Configuration%20Design%20Document%2F03%20Issued%20For%20Approval&FolderCTID=0x0120006CFDFB756210D740899B133F8C129BBB&View={3F451D94-EE87-4C65-B753-5E83FB6E03F4}&InitialTabId=Ribbon%2EDocument&VisibilityContext=WSSTabPersistence](https://mnscorp.sharepoint.com/sites/GM4ProgrammeTeamsite/02_MP/Forms/AllItems.aspx?RootFolder=%2Fsites%2FGM4ProgrammeTeamsite%2F02_MP%2F03%20Detailed%20Design%2F01%20Deliverables%2FMPDD42%20-%20MP%20Technical%20Configuration%20Design%20Document%2F03%20Issued%20For%20Approval&FolderCTID=0x0120006CFDFB756210D740899B133F8C129BBB&View=%7b3F451D94-EE87-4C65-B753-5E83FB6E03F4%7d&InitialTabId=Ribbon%2EDocument&VisibilityContext=WSSTabPersistence)

## Framework Components

Framework components include:

* Configuration file

See section [4.3.1](#_Configuration_files/tables) for more details.

* Event Logger

See section [4.4](#_Event_Logger) for more details.

* Error Handler

See section [4.6](#_Error_Handler) for more details.

* Housekeeper

See section [4.8](#_Housekeeper) for more details.

* Process Scheduler

See section [4.3](#_Process_Scheduler_(Windows)) for more details.

## Core Application Components

Core Application Components that the solution framework uses or supports include:

### EP Tables

* Interface Tables (Master and Transactional Data).
* All the tables under EKB/EXT Schema
* All the tables under Quartz Cubes which will be defined in the Technical configuration (E0259).

In JDA Enterprise Planning all the tables are managed internally by the system. Any other

ways of updating/deleting/adding records to any of the system tables except the

interface tables are not supported.

### Processes

These are the processes internal to EKB and will be executed from wrapper scripts with appropriate parameters based upon the EKB/EP configuration document via a scheduler- control M to process the data.

* seqDBMASTER\_EXT\_TO\_EPServerData(Structure Data Import data stage Job using)
* Pkbfstrupd
* Pkbfdatupd
* pkbfstrstime
* Pkbfdatrm
* Pkbfdatexp
* Cimsyncpkbf
* Cimdatrm

### Services

These are the EP Mid-Tier services and will be stopped/started weekly from wrapper scripts on a weekly schedule via control-M.

* JDABIMessageLog
* JDAEKBStructureService
* JDAEKBStructureCacheService
* JDAEKBStructureFactsSynchronization
* JADEAdminServer
* JDABIApplicationServer

# Physical Design

## Overview

The diagram below in Figure 4 gives the high level overview of the framework.

Windows

JDA Enterprise

Planning

Planning

Oracle

Interface

Tables

Framework

Procedures

Framework

Scripts

RICEF

Procedures

RICEF

Scripts

Application

Process

Application

Tables

Application

Services

Figure 4 Physical Overview

The components can be broadly classified under Windows and Oracle components:

* Windows Components

1. Application Process: These comprise the JDA processes and the utilities which will be invoked by RICEF (Wrapper) scripts.
2. Framework Scripts: These are the generic scripts which will be invoked by the RICEF (Wrapper) scripts. They will have common functionalities that can be used across the RICEF scripts. Eg. Parameters and variables that need to be set
3. RICEF Scripts: These scripts are the wrapper scripts which will invoke the JDA utilities/processes.
4. Application Services: These consist of the JDA Mid Tier/Application Services.

* Oracle Components

1. Interface Tables: These tables will hold the structure and fact data which will be interfaced from the source systems. Interfaces will write the data directly into these tables.
2. Framework Procedures: These are any common routines that will be used by the wrapper scripts. eg. Common utilities that may be needed for error handling.
3. RICEF Procedures: These are any database routines that will need to be created to support the wrapper scripts. eg. A calendar load function that identifies the maximum sequence number from the relation table that will be used while inserting new records into the relation table.
4. Application Tables: These are the internal JDA tables including the foundation tables and quartz cubes. The inbound interfaces do not interact with these tables and are used by the JDA utilities and processes.

## File Systems & Schemas

### Subversion Folder Structure

The following structure will be used in subversion to manage the configuration of development artefacts:

<https://mshsvn/GM/JDA/EnterprisePlanning>

JDA (Folder)

* EnterprisePlanning
  + **Trunk**

- RICEF Objects

* + **Branches**
  + **Tags**

### Windows File System

The below folders will be created on the secondary mid–tier EP servers (primary and fall back server) where all the scripts (non-automation) for RICEF will reside. There will be multiple mid tier servers. The scripts (non-automation) will be deployed in all mid-tier servers which will also be used as the batch servers. The log files for the scripts will be created in a shared location on the primary mid-tier server.

Files will be located as per the below structure

|  |  |  |
| --- | --- | --- |
| Sl. No. | Folder Description | Folder Path |
| 1 | Main batch Folder | D:\batch |
| 2 | Batch Scripts(.bat files) | D:\batch\scripts |
| 3 | Oracle Sql scripts(.sql) | D:\batch\sql |
| 4 | Control file for sql loader(.ctl) | D:\batch\ctl |
| 5 | Main Data folder | D:\batch\data |
| 6 | Inbound data files to JDA EKB | D:\batch\data\inbound |
| 7 | Outbound data files from JDA EKB | D:\batch\data\outbound |
| 8 | Archive data folder | D:\batch\data\archivedata |
| 9 | Xml files for data Export | D:\batch\ xml |
| 10 | Error data files | D:\batch\Errordata |
| 11 | Archive Error data files | D:\batch\Errordata\archiveerror |
| 12 | Log files of batch | D:\batch\logs |
| 13 | Archive log folder | D:\batch\logs\archivelog |
| 14 | Environment/configuration file(<%computername%\_setEnv.bat>) | D:\batch |
| 15 | Automation scripts | <Fileservername>\<Environment\_Name>\Admin\system\util\ |

The automation scripts will reside in the File server which will be developed by JDA automation developer.

<Fileservername>\<Environment\_Name>\Admin\system\util\

### Oracle Schema

* A database schema EXT will be used for holding all necessary AG tables for interfaces for Structures Master data.
* A database schema EKB will be used for holding all necessary TEMP tables and live table for interfaces for Fact Transactional data.
* A database schema MNSI will be created for creating any customized data objects (Procedures, packages, Tables etc) to support Interfaces, enhancements, Conversions and Framework. This schema will refer to the some tables in EXT and EKB schema using the select grants.
* A database schema will be created for each cube which will be defined in the Technical configuration (E0259).
* Primary Key and Foreign Key relationships will be established on the tables as applicable.
* Appropriate indexes, constraints, partitions will be created on the tables as necessary to ensure optimal performance and functionality.
* Database statistics will be utilized where optimum system performance is critical. This will be both scheduled and included in individual batch process where extreme differences in data pre and post operation would significantly affect performance.

### Batch operations

The following link will contain the E2E day-in-the-life (DILO) schedule document for all RICEF objects in scope for MP. Interdependencies, volume, and frequency of the jobs can also be found in the document under this link.

[https://mnscorp.sharepoint.com/sites/GM4ProgrammeTeamsite/02\_MP/Forms/AllItems.aspx?RootFolder=%2Fsites%2FGM4ProgrammeTeamsite%2F02\_MP%2F04%20Build%2F01%20Deliverables%2FMPB01%20-%20Batch%20Schedule%2F03%20Issued%20For%20Approval&FolderCTID=0x0120006CFDFB756210D740899B133F8C129BBB&View={3F451D94-EE87-4C65-B753-5E83FB6E03F4}&InitialTabId=Ribbon%2EDocument&VisibilityContext=WSSTabPersistence](https://mnscorp.sharepoint.com/sites/GM4ProgrammeTeamsite/02_MP/Forms/AllItems.aspx?RootFolder=%2Fsites%2FGM4ProgrammeTeamsite%2F02_MP%2F04%20Build%2F01%20Deliverables%2FMPB01%20-%20Batch%20Schedule%2F03%20Issued%20For%20Approval&FolderCTID=0x0120006CFDFB756210D740899B133F8C129BBB&View=%7b3F451D94-EE87-4C65-B753-5E83FB6E03F4%7d&InitialTabId=Ribbon%2EDocument&VisibilityContext=WSSTabPersistence)

#### Standard Return Codes

Use a pre-defined set of return codes to identify successful and failed scenarios

* **Zero (0)** return code is used for successful scenario.
* **One (1)** used for failed scenario – the process abends (stops).
* **Three (3)** return code for warning scenario. The process will continue but logs something occurred outside of expected norms

#### Standard file/table operation codes

* **Structure Data:** For add/delete/update operation, column (actioncode) in the appropriate interface table will be updated with the flag (1=Add, 2=Update, 3=Delete.) and the records will be added/updated/deleted appropriately.
* **Transactional Data:** There are no such operation codes (for insert, update or delete) for transactional data. The update type of transactional data is a measure specific configuration and is defined in the technical configuration document.

#### Configuration files

* <%computername%\_setEnv.bat> in folder D:\batch will contain the entire environment variable and any other global variables that can be referred and this file will be called in every bat script to set the variables for the proper or expected configuration.
* Hard coding is not allowed in any batch file/SQLs. Parameters will be defined in configuration file.
* Purge criteria for tables will be defined in the technical config document.

## Process Scheduler (Windows)

M&S uses Control-M as its scheduler and an agent will be installed on the JDA secondary mid tier servers where the wrapper scripts (non-automation) will reside.

For automation scripts, Control-M will be running the scripts which reside in the file server. For the batch user (Y-account) to see and execute the scripts in the file server, the folder containing the scripts will be shared and the batch user (Y-account) that the agent uses would be a domain user and added to the share to the file server.

Control –M supports:

* time based scheduling e.g. run a process at 10 pm every night
* day based scheduling e.g. run a process Monday through Friday
* dependency based scheduling e.g. run a process after script X and script Y are complete
* variable/flag based scheduling e.g. schedule conditionally

Control-M will be monitored by SCOM and note when a process abends or has warnings, which will trigger an automatic notification to Netcool generating a REMEDY ticket.

## Auditing and Logging

Auditing and logging for the MP solution is distributed across various application components. Auditing and logs will contain various levels of detail.

Trace files provide summary auditing (at the transaction level) as well as detail level logging (at the row level if there errors or warnings generated)

### Distributed Logs

will be such as <script name>\_<parameter>\_YYYYMMDDHHMM.log

#### Audit logs

Audit logs contains summary batch result and includes the following

1. **Trace File**

All JDA EKB internal server process will create a trace file and that will be redirected to the D:\batch\logs folder by using the TRACE parameter with the naming convention <script name>\_<parameter>\_YYYYMMDDHHMM.trc.This file will contain the detailed information such as timestamp of the utility started, timestamp of utility ended, parameters used to run the utility, rows processed, rows rejected, rows skipped due to warnings ,status stating: process completed successful or process completed with errors etc.

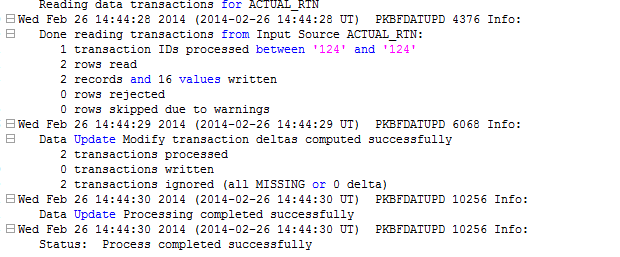


Figure 5 Pkbfdatupd.trc

1. **Pkbf.log**

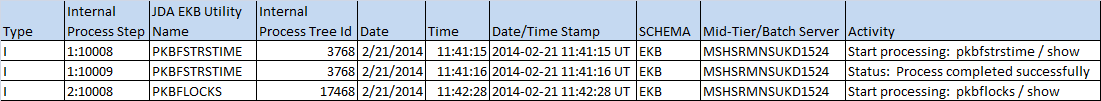


Figure 6 Pkbf.log

1. **Cim.log**

All JDA EKB internal server processes will create a common log files (cim.log) for cube processes in the default JDA configured path which will be decided during the configuration. These are the log files JDA support will look for when working on any EKB issues. These log files will contain the high level information such as utility name ,timestamp of the utility started, timestamp of utility ended, parameters used to run the utility, ran on which server, activity stating: process completed successfully or process completed with errors etc.

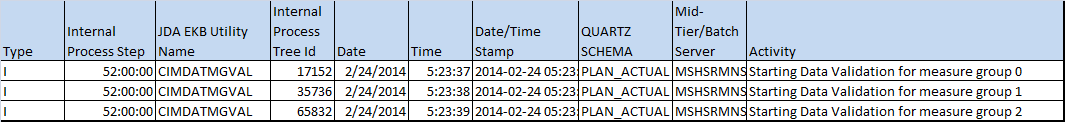
****

Figure 7 Cim.log

1. **Wrapper log**

* If the process is successfully completed, it should generate a return code = 0 and then all process success information will be written to the log files created in the D:\batch\logs folder with the naming convention as <script name>\_<parameter>\_YYYYMMDDHHMM.log.

1. **Automation log**

* executed by the batch user(Y user)
* For example, the batch process for building the user plans batch could be named **BldUser**, and the log file for the batch process will be blduser<Datestamp>.log

#### Detailed Logs

This will contain detailed information whenever there are reported errors (for both master date and transaction data).

1. **Structure Data logs**

The warnings/errors records for Structure data are captured in the table EXT\_EPSD\_ERRORS in EXT schema and EXTMemberErrors.csv file which will be moved in the errordata shared folder as EXTMemberErrors\_YYYYMMDDHHMISS.csv and records associated with the errors stay in their respective interface table which will be spooled in the <Table name\_YYYYMMDDHHMISS.csv> files in errordata shared folder.

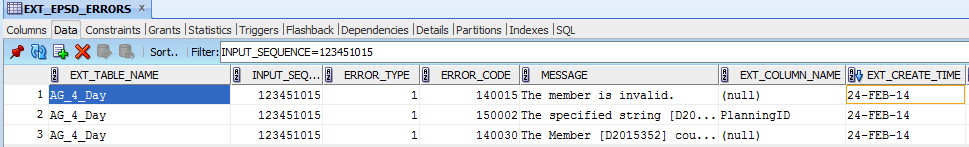


Figure 8 Master data Error table in EXT schema

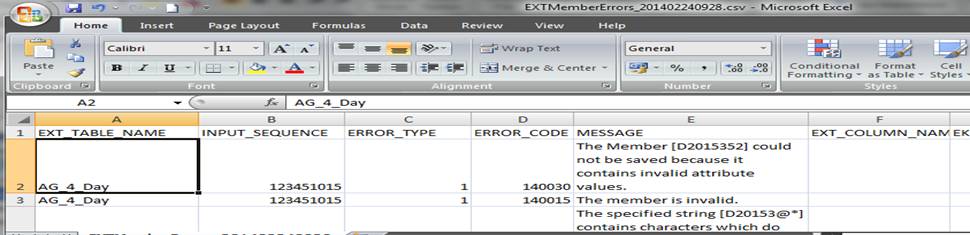


Figure 9 JI export in to ExtMemberErrors .csv

1. **Transactional Data logs**

The warnings/errors for Fact data are captured in the respective PF\_DU\_<FTF Name>\_ERR in EKB schema and records associated with the errors are not moved from their respective interface table which will be spooled in the <Tablename\_YYYYMMDDHHMISS.csv> files in errordata shared folder.

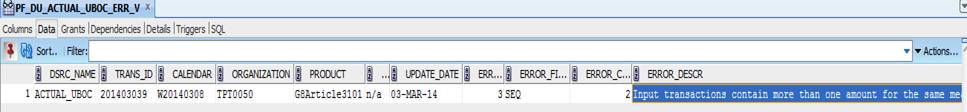


Figure 10 Transaction Error view in EKB schema

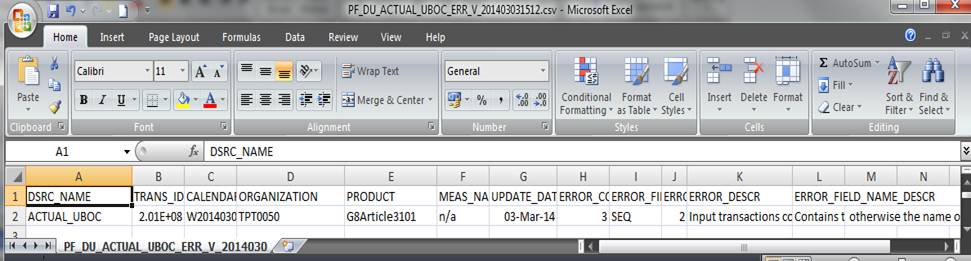


Figure 11 Export of Transaction Error view in to csv file

NOTE: Unless otherwise noted in individual Functional Specifications (FSpecs) OR Technical Configuration OR DILO, if an individual batch process fails due to a hard error the entire nightly batch process should be stopped and an alert should be sent to the nominated business group the Batch process has not completed.

Control-M will receive all batch script return codes (0, 1 and 3) and notify SCOM where the code = 1 or 3. SCOM will then notify Netcool of the error which will create a Remedy ticket and send an email to the nominated representative for that process group stating the related process name, the warning/failed message, and to check the log and error data tables/ files.

### Oracle Logs

* Capture any general oracle error in an exception block and log the appropriate error in the related <scriptname<\_parameter>>\_YYYYMMDDHHMM.log as detailed above in windows logs.

## Data Retention\Replay

### Data Retention Approach

* For JDA interface tables, data will be retained until the following week processing of new master and transactional data.. If the batch is delayed and the records in the staging tables are not processed, therefore the job that is normally purging JDA staging tables needs to suspended by the production support team. After the successful processing of the records the purging job should be started to purge the records. The processing, archiving and purging is configured in control-m.
* Basically all AG\_<hierarchy name> interface table and error table EXT\_EPSD\_ERROR for Structure and all PKB\_ACTUAL\_<FTF> interface table for Fact tables will be truncated in Friday daily batch.
* The error table for fact data PF\_DU\_<FTF Name>\_ERR will be overwritten in the next run.
* The batch which moves the data from interface table to live table and if any error records will be copied from the interface table and also the error records from the error table to the file named as <Table name\_YYMMDDHHMISS.csv> for transactional data and EXTMemberErrors\_ YYMMDDHHMISS.csv for master data in shared errordata folder .These files will be archived in archiveerror folder and will be retained till 30 days as mentioned in housekeeping section [4.8](#_Housekeeper).These csv files will be shared by support team to the respective team for further analaysis.
* For JDA live tables in EP, data retention approach is governed by E0259 (EP Technical Configuration).

### Data Replay Approach

* If the messages are lost from the source, the JDA interface should be able to replay lost messages without any changes to interface.
* For the error records which will be stored in the <Table name\_YYMMDDHHMISS.csv> for transactional data and EXTMemberErrors\_ YYMMDDHHMISS.csv for master data in shared errordata folder will be shared by support team to the respective team which will analyze the records to determine the cause of the error and ask the team to send the corrected data again.
* Once the data is sent again by source in the interface table for structure, the process which moves the data needs to run the same batch job (which internally calls the Data stage job) again and the following processes.
* Once the data is sent again by source in the interface table for Fact, the process which moves data need to run the same batch job again and the following processes.

## Error Handler

If the server processes, other than data stage (loading structure data from interface to fact) fails due to errors other than data errors then the process should be started with the parameter RESTART=Y.

The following examples of errors can be anticipated to occur and will be managed by the JDA EP internal process or batch according to the stated logging and monitoring processes:

* Data inconsistent with expected data type
* Valid date format
* SQL Loader error log, bad file and discard file should be created.
* Oracle error SQLCODE returned
* Null values in table columns
* Numeric overflow
* Duplicates

As a general rule, unless an FSpec specifies otherwise, when a failure occurs the process should continue to the next record until completion. These failed records will be captured and the conditions of their failure recorded as much as possible and placed into a dedicated error table and to the file.

## Report Generator

### User Notifications

The TIVOLI and SCOM systems will monitor the operating system event logs in addition to the Control-M processes. When an error event is called a Remedy ticket will be automatically raised and the appropriate logs and error data will be gathered manually by support team to send to the nominated business representatives.

These events should cover, but are not limited to:

* Errors from logs, txt files.
* System available/unavailable messages to users after nightly batch processing or critical system failures.

### Exception Reports (Warnings)

The EP internal processes generate various auditing, tracing, logs and other warnings when the situations occur and the associated records and logs should be picked up by SCOM and sent to the nominated business representative for review and resolution.

## Housekeeper

General housekeeping rules should apply.

* Every week, one housekeeping script before the weekly batch will take all files from the previous weeks in the logs, data (inbound and outbound) directories and in errordata folder and then will be zipped into a single file and placed into the archive directory of the respective folder (i.e. archivedata for data, archivelog for logs, archiveerror for errordata) before the daily/ weekly batch processing begins. These files (logs , data or errordata) should be called BatchRun\_{<logs>or<data>or<errordata>}\_YYYYMMDD.zip

Example: If the batch runs on weekly and the current date is 20130614 then the zip file will be created in their respective archive directories as below:

BatchRun\_logs\_20130614.zip

BatchRun\_data\_20130614.zip

BatchRun\_errordata\_20130614.zip

* Every week one housekeeping script before the weekly batch run will purge the zip file BatchRun\_{<logs> }\_YYYYMMDDHHMM.zip , BatchRun\_{<data> }\_YYYYMMDDHHMM.zip and BatchRun\_{<logs> }\_YYYYMMDDHHMM.zip from archive folders(archivedata for data, archivelog for logs, archiveerror for errordata) older than 30 days.
* On Friday night after the Daily run, one housekeeping batch will truncate AG\_<hierarchy name> interface tables and error table EXT\_EPSD\_ERROR for Structure and all PKB\_ACTUAL\_<FTF> interface tables and PF\_DU\_<FTF Name>\_ERR for Fact tables.
* Other EKB logs i.e. Quartz cube logs and EKBF logs will also be included in the house keeping process for batches. These logs will be archived into the same location as the batch logs and will be retained for 30 days.
* Mid Tier logs will be retained for 1 week in the respective Mid tier servers.

## Reusable Routines

* Create a single batch script with parameters, wherever needed, so that the same batch script can be used for multiple purposes based on parameters.

For EX: There will be one batch script for Pkbfdataupd process which will be used for loading sales data for different fact table family

# Back-Ups

## Oracle Backups

* Schedule initial and incremental daily backups of Oracle databases before and after the batch and the script will be given by the E-DBA and run through control-M.

## Windows Backups

* Daily and weekly backup of user and admin plan files on file server will be done before and after the batch.
* Mid-tier Backup will be done daily.
* These will be done by the backup team as a part of backup strategy.

------------------------------------------- End of the Document ------------------------------------------------