

PlanNOW!

Interface Design Document

Enterprise Planning Server Edition Release 2020.1

**Date Created**: March 31, 2022

**Date Updated**: March 31, 2022

**Version**: 1.0

Legal Notice

Copyright © 2022, Blue Yonder, Inc. All rights reserved. Blue Yonder is a Registered Trademark of Blue Yonder, Inc. All other company and product names may be Trademarks, Registered Trademarks or Service Marks of the companies with which they   
are associated. Blue Yonder reserves the right at any time and without notice to change these materials or any of the functions, features or specifications of any of the software described herein. Blue Yonder shall have no warranty obligation with respect   
to these materials or the software described herein, except as approved in Blue Yonder’s Software License Agreement with an authorized licensee.

Table of Contents

[Interface Design Document 5](#_Toc98943786)

[Document Purpose 5](#_Toc98943787)

[Audience 5](#_Toc98943788)

[Revision History 5](#_Toc98943789)

[Section 1: EKB Integration 6](#_Toc98943790)

[Directory Structure 6](#_Toc98943791)

[Section 2: Structure Data 7](#_Toc98943792)

[Product Member Data 7](#_Toc98943793)

[Organization Member Data 9](#_Toc98943794)

[Calendar Member Data 10](#_Toc98943795)

[Parent/Child Relationship Meta Data 12](#_Toc98943796)

[User Defined Pick List Meta Data 13](#_Toc98943797)

[Section 3: Fact Data 14](#_Toc98943798)

[Section 4: Interface Data and Control Files 16](#_Toc98943799)

[Product Member Data Files 17](#_Toc98943800)

[Organization Member Data Files 18](#_Toc98943801)

[Calendar Member Data Files 19](#_Toc98943802)

[Structure Parent/Child Relationship Meta Data File 19](#_Toc98943803)

[Structure Pick List Meta Data File 19](#_Toc98943804)

[Structure Data Control File 20](#_Toc98943805)

[Fact Data Files 21](#_Toc98943806)

[Fact Data Control File 22](#_Toc98943807)

[Section 5: Error Reporting 23](#_Toc98943808)

[Oracle SQLLDR Processing 23](#_Toc98943809)

[JDA Integrator and Foundation Processing 24](#_Toc98943810)

[Appendix A 26](#_Toc98943811)

[Supporting Documents 26](#_Toc98943812)

# Interface Design Document

### Document Purpose

The purpose of the EP Interface Design Document is to define how to accomplish structure and fact data integration into the Enterprise Knowledge Base (EKB).

Refer to the EP Technical Design Document for specifics about configuration of EKB and refer to the EP Product documentation for details about the EKB database.

### Audience

This document is directed at EP Technical personnel who have some knowledge of the EKB database and concepts related to data extracting, transforming and loading (ETL).

### Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 1.0 | 03-03-2022 |  | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Section 1: EKB Integration

EKB integration is performed by creating specially formatted and named data files that map to staging tables within the EKB database. This document defines file formats, file names and the supporting file system directory structure. The directory structure is used for depositing data files and retrieving any files indicating data file processing errors.

### Directory Structure

The following directories are usually contained within a parent directory called EP or SFTP depending on the implementation (See the SFTPPATH environment variable in the Batch Design Document for the exact location).

These are the directories contained within the SFTPPATH:

|  |  |
| --- | --- |
| **DIRECTORY NAME** | **DESCRIPTION** |
| Fact\_History | Used for fact data history loads only and processed by a specialized job intended for loading large quantities of data. |
| Inbox | Used for loading structure and fact data delta loads. Processed by a scheduler controlled job and typically on a weekly basis. |
| Inbox\Archive | Processed Structure and Fact data files with control files are archived here. Files kept for 28 days. |
| Outbox | Location where system generated EKB extracts can be found. |
| SQLLDR\_Errors | Location where SQL Loader errors from data processed in Fact\_History or Inbox directories can be found. |
| SQLLDR\_Logs | Location where SQL Loader output from data processed in Fact\_History or Inbox directories can be found. |
| Structure\_Load\_Errors | Location where EP errors from structure data processed in Fact\_History or Inbox directories can be found. |
| TY\_Load\_Errors | Location where EP errors from fact data processed in Fact\_History or Inbox directories can be found. |

# Section 2: Structure Data

There are 2 types of structure data in EKB: Member Data and Meta Data. Member Data represents hierarchy information while Meta Data supports hierarchy information that defines parent/child relationships and user defined pick lists used in some field validations.

### Product Member Data

|  |  |  |
| --- | --- | --- |
| **BASE FIELDS** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| INPUTSEQUENCE | Oracle NUMBER | A unique numeric sequence that identifies the order in which the records are to be processed. Required for all requests. |
| ACTIONCODE | 1 | One of three values: 1 for Add, 2 for Update and 3 for Delete. When a member is deleted, all data associated with that member will be deleted as well. A member cannot be deleted if it has children. Required for all requests. |
| CURRENT\_ID | 30 | The current ID value stored in EKB. Used for Update and Delete requests. |
| ID | 30 | The member ID. Required for Add requests. Optional for Update requests. Must contain printable ASCII characters without spaces. Unique within the product level. |
| PARENT\_MEMBER\_ID | 30 | The ID of the parent member within the same hierarchy. Required for Add requests. Use the EXT\_EPSD\_IMPORT\_MEMBER\_REL record type to:   * update the parent member id * add, modify or delete the parent member in an alternate hierarchy |
| CULTURE\_NAME | 10 | Set to ‘en-US’. Required for all requests. |
| SHORT\_DESCRIPTION | 50 | A short description of the member. Required for Add and Update requests. |
| LONG\_DESCRIPTION | 300 | An optional more detailed description of the member. |
| PLANNINGID | 18 | The member’s Planning ID. Required for Add and Update requests. Must begin with a letter, no spaces allowed and can contain the underscore character. Unique within the product hierarchy. |

***NOTE****: It is a common practice to use the same value for ID and PLANNINGID.*

|  |  |  |
| --- | --- | --- |
| **IMPLEMENTATION SPECIFIC FIELDS** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| KEYBRAND | 50 | Allowed values are constrained by the KEYBRAND pick list. Pick list maintained using the EXT\_EPSD\_IMPORT\_ATTR\_DEF\_POT request. |
| LIFESTYLE | 50 | Allowed values are constrained by the LIFESTYLE pick list. Pick list maintained using the EXT\_EPSD\_IMPORT\_ATTR\_DEF\_POT request. |

***NOTE****: For simplicity of interface logic, it is good practice to supply values for all fields regardless of ACTIONCODE especially in the case of Adds and Updates.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MEMBER ID FORMATS** | | | | |
| **LEVEL** | **ID/PLANNING ID** | **SAMPLE** | **SHORT\_DESCRIPTION** | **MEMBER COUNT** |
| P1 |  |  |  |  |
| P2 |  |  |  |  |
| TOTPROD |  |  |  |  |
| DIV |  |  |  |  |
| DPT |  |  |  |  |
| SUBDPT |  |  |  |  |
| CLASS |  |  |  |  |
| SUBCLASS |  |  |  |  |
| STYLE |  |  |  |  |
| STYLECOLOR |  |  |  |  |

### Organization Member Data

|  |  |  |
| --- | --- | --- |
| **BASE FIELDS** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| INPUTSEQUENCE | Oracle NUMBER | A unique numeric sequence that identifies the order in which the records are to be processed. Required for all requests. |
| ACTIONCODE | 1 | One of three values: 1 for Add, 2 for Update and 3 for Delete. When a member is deleted, all data associated with that member will be deleted as well. A member cannot be deleted if it has children. Required for all requests. |
| CURRENT\_ID | 30 | The current ID value stored in EKB. Used for Update and Delete requests. |
| ID | 30 | The member ID. Required for Add requests. Optional for Update requests. Unique within the organization level. Must contain printable ASCII characters without spaces. |
| PARENT\_MEMBER\_ID | 30 | The ID of the parent member within the same hierarchy. Required for Add requests. Use the EXT\_EPSD\_IMPORT\_MEMBER\_REL record type to:   * update the parent member id within a hierarchy * add, modify or delete the parent member across hierarchies |
| CULTURE\_NAME | 10 | Set to ‘en-US’. Required for all requests. |
| SHORT\_DESCRIPTION | 50 | A short description of the member. Required for Add and Update requests. |
| LONG\_DESCRIPTION | 300 | An optional more detailed description of the member. |
| PLANNINGID | 18 | The member’s Planning ID. Required for Add and Update requests. Must begin with a letter, no spaces allowed and can contain the underscore character. Unique within the product hierarchy. |

***NOTE****: It is a common practice to use the same value for ID and PLANNINGID.*

|  |  |  |
| --- | --- | --- |
| **IMPLEMENTATION SPECIFIC FIELDS** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
|  |  |  |
|  |  |  |

***NOTE****: For simplicity of interface logic, it is good practice to supply values for all fields regardless of ACTIONCODE especially in the case of Adds and Updates.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MEMBER ID FORMATS** | | | | |
| **LEVEL** | **ID/PLANNING ID** | **SAMPLE** | **SHORT\_DESCRIPTION** | **MEMBER COUNT** |
| P1 |  |  |  |  |
| TOTORG |  |  |  |  |
| BU |  |  |  |  |
| CHANNEL |  |  |  |  |
| REGION |  |  |  |  |
| DISTRICT |  |  |  |  |
| TOUCHPOINT |  |  |  |  |

### Calendar Member Data

PlanNOW uses a 4-4-5 retail planning calendar with the fiscal year starting in February. For the initial implementation, the years designated as 53 week years include (2023).

|  |  |  |
| --- | --- | --- |
| **BASE FIELDS** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| INPUTSEQUENCE | Oracle NUMBER | A unique numeric sequence that identifies the order in which the records are to be processed. Required for all requests. |
| ACTIONCODE | 1 | One of two values: 1 for Add and 2 for Update. Calendar members cannot be deleted. Required for all requests. |
| CURRENT\_ID | 30 | The current ID value stored in EKB. Used for Update requests. |
| ID | 30 | The member ID. Required for Add requests. Optional for Update requests. Unique within the calendar level. Must contain printable ASCII characters without spaces. |
| PARENT\_MEMBER\_ID | 30 | The ID of the parent member within the same hierarchy. Required for Add requests. Unused by Update requests. Use the EXT\_EPSD\_IMPORT\_MEMBER\_REL record type to:   * update the parent member id * add or modify the parent member in an alternate hierarchy |
| CULTURE\_NAME | 10 | Set to ‘en-US’. Required for Add requests. |
| SHORT\_DESCRIPTION | 50 | A short description of the member. Required for Add and Update requests. |
| LONG\_DESCRIPTION | 300 | An optional more detailed description of the member. |
| PLANNINGID | 18 | The member’s Planning ID. Required for Add and Update requests. Must begin with a letter, no spaces allowed and can contain the underscore character. Unique within the product hierarchy. |
| CAL\_STARTDATE | 10 | The beginning date of the Calendar Member in the format ‘MM-DD-YYYY’. Required for Add requests. Unused by Update requests. Value of this field is automatically maintained and could change based on the reclassing of a child member. |

***NOTE****: It is a common practice to use the same value for ID and PLANNINGID.*

|  |  |  |
| --- | --- | --- |
| **IMPLEMENTATION SPECIFIC FIELDS** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
|  |  |  |
|  |  |  |

***NOTE****: For simplicity of interface logic, it is good practice to supply values for all fields regardless of ACTIONCODE especially in the case of Adds and Updates.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MEMBER ID FORMATS** | | | | |
| **LEVEL** | **ID/PLANNING ID** | **SAMPLE** | **SHORT\_DESCRIPTION** | **MEMBER COUNT** |
| YEAR | FY[yyyy] | FY2022 | FY [yyyy] | 1 |
| SEASON | S[yyyy]{0} | S20221 | [season] - [yy] | 2 |
| QUARTER | Q[yyyy]{0} | Q20221 | Qtr {0} - [yy] | 4 |
| MONTH | M[yyyy]{0:00} | M202201 | Mon {0:00} - [yy] | 12 |
| WEEK | W[yyyy]{0:00} | W202201 | Wk {0:00} - [yy] | 52/53 |
| DAY | D[yyyy]{0:000} | D2022001 | Day {0:000} - [yy] | 365/371 |

### Parent/Child Relationship Meta Data

The EXT\_EPSD\_IMPORT\_MEMBER\_REL table is used for the following:

* Modifying a Member’s parent relationship within the Main Hierarchy
* Establishing or modifying a Member’s parent relationship to an Alternate Hierarchy

Modifying a Member’s parent relationship is referred to as “reclassing”. The following chart shows the columns within the EXT\_EPSD\_IMPORT\_MEMBER\_REL table…

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| INPUTSEQUENCE | Oracle NUMBER | A unique numeric sequence that identifies the order in which the records are to be processed. Required for all requests. |
| ACTIONCODE | 1 | One of three values: 1 for Add, 2 for Update and 3 for Delete. Only update is supported for changing parentage within a hierarchy. All three values are supported for manipulating parentage across hierarchies. Required for all requests. |
| BUS\_ENTITY\_TECH\_KEY | 1 | One of three values: 1 for Product, 2 for Organization and 4 for Calendar. Required for all requests. |
| CHILD\_MEMBER\_ID | 30 | The member id of the member being reclassed. Required for all requests. |
| CHILD\_LEVEL\_ID | 30 | The level of the member being reclassed. Required for all requests. |
| PARENT\_MEMBER\_ID | 30 | The grouping member being reclassed to. Required for all requests. |
| PARENT\_LEVEL\_ID | 30 | The level of the grouping member being reclassed to. Required for all requests. |

***NOTE****: For simplicity of interface logic, it is good practice to supply values for all fields regardless of ACTIONCODE especially in the case of Adds and Updates.*

### User Defined Pick List Meta Data

The EXT\_EPSD\_IMPORT\_ATTR\_DEF\_POT table is used to modify the list of possible values for an Attribute.

The following chart shows the columns within the EXT\_EPSD\_IMPORT\_ATTR\_DEF\_POT table…

|  |  |  |
| --- | --- | --- |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| INPUTSEQUENCE | Oracle NUMBER | A unique numeric sequence that identifies the order in which the records are to be processed. Required for all requests. |
| ACTIONCODE | 1 | One of three values: 1 for Add, 2 for Update and 3 for Delete. Required for all requests. |
| CURRENT\_ID | 30 | The ID of the Attribute the pick list value is being added for. Required for all requests. |
| ATTR\_TYPE | 2 | Set to ‘6’. Required for all requests. |
| CURRENT\_VALUE\_ID | 30 | The ID of the pick list value stored in EKB. Required for Update and Delete. |
| PROCESS\_VALUE\_ID | 1 | ‘T’ or ‘F’. Indicates the value id should be updated in EKB. Used for Add and Update only. |
| VALUE\_ID | 30 | The new ID of the pick list value. Used for Add and Update only. |
| CULTURE\_NAME | 10 | Set to ‘en-US’. Required for all requests. |
| PROCESS\_STRING\_VALUE | 1 | ‘T’ or ‘F’. Indicates the string value should be updated in EKB. Used for Add and Update only. |
| STRING\_VALUE | 50 | The pick list value. Used for Add and Update only. |

***NOTE****: For simplicity of interface logic, it is good practice to supply values for all fields regardless of ACTIONCODE especially in the case of Adds and Updates.*

# Section 3: Fact Data

|  |  |  |
| --- | --- | --- |
| **BASE FIELDS** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| TRANS\_ID | Oracle NUMBER | A numeric value uniquely identifying the transaction for all records in the data load. Subsequent data loads must have a greater value. It is suggested to use the format *YYYYMMDDHHMM.* Required. |
| PRODUCT\_NAME | 30 | The ID value of the **STYLECOLOR** product member data is being loaded for. Required. |
| ORGANIZATION\_NAME | 30 | The ID value of the **TOUCHPOINT** organization member data is being loaded for. Required. |
| CALENDAR\_NAME | 30 | The ID value of the **WEEK** calendar member data is being loaded for. Required. |

|  |  |  |
| --- | --- | --- |
| **IMPLEMENTATION SPECIFIC FIELDS – PKB\_ACTUAL\_INVBAL** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| TRANS\_TYPE | 1 | <NULL>: The row will be processed using the update method specified for the input source.  ‘S’: The row will be processed using the Restatement method. Note that if one row for a Transaction ID specifies this method, all rows with the same Transaction ID must also specify this method.  ‘M’: The row will be processed using the Modify method.  ‘A’: The row will be processed using the Adjustment method. |
| EOP\_CST | Oracle NUMBER | Ending Inv Cost |
| EOP\_RTL | Oracle NUMBER | Ending Inv Retail |
| EOP\_UNT | Oracle NUMBER | Ending Inv Units |

|  |  |  |
| --- | --- | --- |
| **IMPLEMENTATION SPECIFIC FIELDS – PKB\_ACTUAL\_MKDNS** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| MDPMD\_CST | Oracle NUMBER | Perm Mkdns Cost |
| MDPMD\_RTL | Oracle NUMBER | Perm Mkdns Retail |
| MDPOS\_CST | Oracle NUMBER | POS Mkdns Cost |
| MDPOS\_RTL | Oracle NUMBER | POS Mkdns Retail |

|  |  |  |
| --- | --- | --- |
| **IMPLEMENTATION SPECIFIC FIELDS – PKB\_ACTUAL\_ON\_ORDER** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| OO\_CST | Oracle NUMBER | On Order Cost |
| OO\_RTL | Oracle NUMBER | On Order Retail |
| OO\_UNT | Oracle NUMBER | On Order Units |

|  |  |  |
| --- | --- | --- |
| **IMPLEMENTATION SPECIFIC FIELDS – PKB\_ACTUAL\_OTHINV** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| SHR\_CST | Oracle NUMBER | Shrink Cost |
| SHR\_RTL | Oracle NUMBER | Shrink Retail |
| SHR\_UNT | Oracle NUMBER | Shrink Units |

|  |  |  |
| --- | --- | --- |
| **IMPLEMENTATION SPECIFIC FIELDS – PKB\_ACTUAL\_RECEIPTS** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| RECCOM\_CST | Oracle NUMBER | Receipts Committed Cost |
| RECCOM\_RTL | Oracle NUMBER | Receipts Committed Retail |
| RECCOM\_UNT | Oracle NUMBER | Receipts Committed Units |
| RECNCOM\_CST | Oracle NUMBER | Receipts NonCommitted Cost |
| RECNCOM\_RTL | Oracle NUMBER | Receipts NonCommitted Retail |
| RECNCOM\_UNT | Oracle NUMBER | Receipts NonCommitted Units |

|  |  |  |
| --- | --- | --- |
| **IMPLEMENTATION SPECIFIC FIELDS – PKB\_ACTUAL\_SALES** | | |
| **FIELD NAME** | **MAX LENGTH** | **DESCRIPTION** |
| SLS\_CST | Oracle NUMBER | Sales Cost |
| SLS\_RTL | Oracle NUMBER | Sales Retail |
| SLS\_UNT | Oracle NUMBER | Sales Unit |

# Section 4: Interface Data and Control Files

Interface files must be written to the Inbox directory and are of two types: data files and control files. Each load is comprised of a single control file and one or more data files. Data files are written first followed by the control file. The existence of the control file indicates a ready status for processing to begin.

It is advised that all data files and the control file contain the same timestamp prefix in their file names for a single load. In addition, it is also advised that fact data files use a TRANS\_ID value the same as the file name timestamp. Each load should have a unique and progressing timestamp. Control files are processed in sorted order. Data files are processed in the order listed in the control file.

Data files contain information destined for the named table in the EKB database whereas the control file identifies all the data files belonging to a load.

Records within the data files contain positional field values separated by pipe (‘|’) characters. There are no pipe characters at the beginning or end of the record. A carriage return is at the end of each record. Each data file begins with a static header record.

### Product Member Data Files

|  |  |  |
| --- | --- | --- |
| **LEVEL NAME** | **DATA FILE NAME** | **DATA FILE HEADER AND RECORD FORMAT** |
| P1 | *<YYYYMMDDHHMM>*\_AG\_1\_P1.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| P2 | *<YYYYMMDDHHMM>*\_AG\_1\_P2.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| TOTPROD | *<YYYYMMDDHHMM>*\_AG\_1\_TOTPROD.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| DIV | *<YYYYMMDDHHMM>*\_AG\_1\_DIV.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| DPT | *<YYYYMMDDHHMM>*\_AG\_1\_DPT.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| SUBDPT | *<YYYYMMDDHHMM>*\_AG\_1\_SUBDPT.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| CLASS | *<YYYYMMDDHHMM>*\_AG\_1\_CLASS.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| SUBCLASS | *<YYYYMMDDHHMM>*\_AG\_1\_SUBCLASS.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| STYLE | *<YYYYMMDDHHMM>*\_AG\_1\_STYLE.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| STYLECOLOR | *<YYYYMMDDHHMM>*\_AG\_1\_STYLECOLOR.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID|**KEYBRAND|LIFESTYLE** |

***NOTE****: Data files need to be zipped before depositing into the Inbox directory. Empty .DAT files are optional unless referenced by the control file.*

### Organization Member Data Files

|  |  |  |
| --- | --- | --- |
| **LEVEL NAME** | **DATA FILE NAME** | **DATA FILE HEADER AND RECORD FORMAT** |
| P1 | *<YYYYMMDDHHMM>*\_AG\_2\_P1.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| TOTORG | *<YYYYMMDDHHMM>*\_AG\_2\_TOTORG.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| BU | *<YYYYMMDDHHMM>*\_AG\_2\_BU.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| CHANNEL | *<YYYYMMDDHHMM>*\_AG\_2\_CHANNEL.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| REGION | *<YYYYMMDDHHMM>*\_AG\_2\_REGION.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| DISTRICT | *<YYYYMMDDHHMM>*\_AG\_2\_DISTRICT.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |
| TOUCHPOINT | *<YYYYMMDDHHMM>*\_AG\_2\_TOUCHPOINT.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID |

***NOTE****: Data files need to be zipped before depositing into the Inbox directory. Empty .DAT files are optional unless referenced by the control file.*

### Calendar Member Data Files

|  |  |  |
| --- | --- | --- |
| **LEVEL NAME** | **DATA FILE NAME** | **DATA FILE HEADER AND RECORD FORMAT** |
| TOTYEAR | *<YYYYMMDDHHMM>*\_AG\_4\_TOTYEAR.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID|CAL\_STARTDATE |
| YEAR | *<YYYYMMDDHHMM>*\_AG\_4\_YEAR.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID|CAL\_STARTDATE |
| SEASON | *<YYYYMMDDHHMM>*\_AG\_4\_SEASON.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID|CAL\_STARTDATE |
| QUARTER | *<YYYYMMDDHHMM>*\_AG\_4\_QUARTER.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID|CAL\_STARTDATE |
| PERIOD | *<YYYYMMDDHHMM>*\_AG\_4\_PERIOD.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID|CAL\_STARTDATE |
| WEEK | *<YYYYMMDDHHMM>*\_AG\_4\_WEEK.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID|CAL\_STARTDATE |
| DAY | *<YYYYMMDDHHMM>*\_AG\_4\_DAY.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ID|  PARENT\_MEMBER\_ID|CULTURE\_NAME|SHORT\_DESCRIPTION|  LONG\_DESCRIPTION|PLANNINGID|CAL\_STARTDATE |

***NOTE****: Data files need to be zipped before depositing into the Inbox directory. Empty .DAT files are optional unless referenced by the control file.*

### Structure Parent/Child Relationship Meta Data File

|  |  |
| --- | --- |
| **DATA FILE NAME** | **DATA FILE HEADER AND RECORD FORMAT** |
| *<YYYYMMDDHHMM>*\_EXT\_EPSD\_IMPORT\_MEMBER\_REL.DAT | INPUTSEQUENCE|ACTIONCODE|BUS\_ENTITY\_TECH\_KEY|  CHILD\_MEMBER\_ID|CHILD\_LEVEL\_ID|PARENT\_MEMBER\_ID|  PARENT\_LEVEL\_ID |

***NOTE****: Data files need to be zipped before depositing into the Inbox directory. Empty .DAT files are optional unless referenced by the control file.*

### Structure Pick List Meta Data File

|  |  |
| --- | --- |
| **DATA FILE NAME** | **DATA FILE HEADER AND RECORD FORMAT** |
| *<YYYYMMDDHHMM>*\_EXT\_EPSD\_IMPORT\_ATTR\_DEF\_POT.DAT | INPUTSEQUENCE|ACTIONCODE|CURRENT\_ID|ATTR\_TYPE|  CURRENT\_VALUE\_ID|PROCESS\_VALUE\_ID|VALUE\_ID|  CULTURE\_NAME|PROCESS\_STRING\_VALUE|STRING\_VALUE |

***NOTE****: Data files need to be zipped before depositing into the Inbox directory. Empty .DAT files are optional unless referenced by the control file.*

### Structure Data Control File

The structure control file lists each file contained in the structure data load. Each record in this file contains the name of a structure data file with its record count separated by a pipe character (‘|’). The record count includes the header. Each record ends in a carriage return.

The batch job that processes the structure control file will wait until it arrives (with a batch configurable timeout value). If for some reason there are no updates to send, an empty control file should be sent to satisfy the batch job.

|  |  |
| --- | --- |
| **CONTROL FILE NAME** | **CONTROL FILE CONTENTS** |
| *<YYYYMMDDHHMM>*\_STRUCTURE.CON | *<YYYYMMDDHHMM>*\_AG\_1\_P1.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_1\_P2.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_1\_TOTPROD.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_1\_DIV.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_1\_DPT.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_1\_SUBDPT.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_1\_CLASS.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_1\_SUBCLASS.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_1\_STYLE.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_1\_STYLECOLOR.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_2\_P1.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_2\_TOTORG.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_2\_BU.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_2\_CHANNEL.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_2\_REGION.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_2\_DISTRICT.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_2\_TOUCHPOINT.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_4\_TOTYEAR.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_4\_YEAR.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_4\_SEASON.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_4\_QUARTER.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_4\_PERIOD.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_4\_WEEK.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_AG\_4\_DAY.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_EXT\_EPSD\_IMPORT\_ATTR\_DEF\_POT.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_EXT\_EPSD\_IMPORT\_MEMBER\_REL.DAT|*[record count]* |

***NOTE****: Only those data files being written to the Inbox need referencing by the control file.*

### Fact Data Files

Each fact data record contains required base attribute values and optional implementation specific fields (measures). **It is highly advised to avoid sending measure values of zero as well as records that contain all null measure values**.

|  |  |  |
| --- | --- | --- |
| **FACT TABLE NAME** | **DATA FILE NAME** | **DATA FILE HEADER AND RECORD FORMAT** |
| ACTUAL\_INVBAL | *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_INVBAL.DAT | TRANS\_ID|PRODUCT\_NAME|ORGANIZATION\_NAME|  CALENDAR\_NAME|**EOP\_CST|EOP\_RTL|EOP\_UNT** |
| ACTUAL\_MKDNS | *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_MKDNS.DAT | TRANS\_ID|PRODUCT\_NAME|ORGANIZATION\_NAME|  CALENDAR\_NAME|**MDPMD\_CST|MDPMD\_RTL|**  **MDPOS\_CST|MDPOS\_RTL** |
| ACTUAL\_ON\_ORDER | *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_ON\_ORDER.DAT | TRANS\_ID|PRODUCT\_NAME|ORGANIZATION\_NAME|  CALENDAR\_NAME|**OO\_CST|OO\_RTL|OO\_UNT** |
| ACTUAL\_OTHINV | *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_OTHINV.DAT | TRANS\_ID|PRODUCT\_NAME|ORGANIZATION\_NAME|  CALENDAR\_NAME|**SHR\_CST|SHR\_RTL|SHR\_UNT** |
| ACTUAL\_RECEIPTS | *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_RECEIPTS.DAT | TRANS\_ID|PRODUCT\_NAME|ORGANIZATION\_NAME|  CALENDAR\_NAME|**RECCOM\_CST|RECCOM\_RTL|**  **RECCOM\_UNT|RECNCOM\_CST|RECNCOM\_RTL|**  **RECNCOM\_UNT** |
| ACTUAL\_SALES | *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_SALES.DAT | TRANS\_ID|PRODUCT\_NAME|ORGANIZATION\_NAME|  CALENDAR\_NAME|**SLS\_CST|SLS\_RTL|SLS\_UNT** |

***NOTE****: Data files need to be zipped before depositing into the Inbox directory. Empty .DAT files are optional unless referenced by the control file.*

### Fact Data Control File

The fact control file lists each file contained in the fact data load. Each record in this file contains the name of a fact data file with its record count separated by a pipe character (‘|’). The record count includes the header. Each record ends in a carriage return.

The batch job that processes the fact control file will wait until it arrives (with a batch configurable timeout value). If for some reason there are no updates to send, an empty control file should be sent to satisfy the batch job.

|  |  |
| --- | --- |
| **CONTROL FILE NAME** | **RECORD FORMAT** |
| *<YYYYMMDDHHMM>*\_FACT.CON | *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_INVBAL.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_MKDNS.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_ON\_ORDER.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_OTHINV.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_RECEIPTS.DAT|*[record count]* |
| *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_SALES.DAT|*[record count]* |

***NOTE****: Only those data files being written to the Inbox need referencing by the control file.*

# Section 5: Error Reporting

The processing of structure and fact data can result in errors at two different stages. The first stage is SQL Loader processing into the staging tables and the second stage is EP processing from the EKB staging tables into the EKB structure member and foundation granite tables.

### Oracle SQLLDR Processing

Rejected data file records from SQL Loader are written to the SQLLDR\_Errors directory and will have the same name as the .DAT file but with the .DAT extension replaced with .BAD.

|  |
| --- |
| **SQLLDR ERROR FILE NAMES** |
| *<YYYYMMDDHHMM>*\_AG\_1\_P1.BAD |
| *<YYYYMMDDHHMM>*\_AG\_1\_P2.BAD |
| *<YYYYMMDDHHMM>*\_AG\_1\_TOTPROD.BAD |
| *<YYYYMMDDHHMM>*\_AG\_1\_DIV.BAD |
| *<YYYYMMDDHHMM>*\_AG\_1\_DPT.BAD |
| *<YYYYMMDDHHMM>*\_AG\_1\_SUBDPT.BAD |
| *<YYYYMMDDHHMM>*\_AG\_1\_CLASS.BAD |
| *<YYYYMMDDHHMM>*\_AG\_1\_SUBCLASS.BAD |
| *<YYYYMMDDHHMM>*\_AG\_1\_STYLE.BAD |
| *<YYYYMMDDHHMM>*\_AG\_1\_STYLECOLOR.BAD |
| *<YYYYMMDDHHMM>*\_AG\_2\_P1.BAD |
| *<YYYYMMDDHHMM>*\_AG\_2\_TOTORG.BAD |
| *<YYYYMMDDHHMM>*\_AG\_2\_BU.BAD |
| *<YYYYMMDDHHMM>*\_AG\_2\_CHANNEL.BAD |
| *<YYYYMMDDHHMM>*\_AG\_2\_REGION.BAD |
| *<YYYYMMDDHHMM>*\_AG\_2\_DISTRICT.BAD |
| *<YYYYMMDDHHMM>*\_AG\_2\_TOUCHPOINTDAT.BAD |
| *<YYYYMMDDHHMM>*\_AG\_4\_TOTYEAR.BAD |
| *<YYYYMMDDHHMM>*\_AG\_4\_YEAR.BAD |
| *<YYYYMMDDHHMM>*\_AG\_4\_SEASON.BAD |
| *<YYYYMMDDHHMM>*\_AG\_4\_QUARTER.BAD |
| *<YYYYMMDDHHMM>*\_AG\_4\_PERIOD.BAD |
| *<YYYYMMDDHHMM>*\_AG\_4\_WEEK.BAD |
| *<YYYYMMDDHHMM>*\_AG\_4\_DAY.BAD |
| *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_INVBAL.BAD |
| *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_MKDNS.BAD |
| *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_ON\_ORDER.BAD |
| *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_OTHINV.BAD |
| *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_RECEIPTS.BAD |
| *<YYYYMMDDHHMM>*\_PKB\_ACTUAL\_SALES.BAD |

### JDA Integrator and Foundation Processing

Once data file records pass SQL Loader validation and have been successfully staged in EKB, then the next step is for structure data to be processed by the JDA Integrator and fact data to be processed by the Foundation Data Update.

**JDA Integrator**

When errors occur during JDA Integrator (JI) processing, an error file is written for each structure data file containing records that failed EP validation. There will also be a dump of the JI EXT\_EPSD\_ERRORS table for more details about why validation failed. The error files are listed within a control file where each record contains the name of the error file and its record count separated by a pipe (‘|’) character.

These files are written to the Structure\_Load\_Errors directory.

|  |  |
| --- | --- |
| **CONTROL FILE NAME** | **CONTROL FILE CONTENTS** |
| *<YYYYMMDDHHMMSS>*\_STRUCTURE.CON | *<YYYYMMDDHHMMSS>*\_AG\_1\_P1.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_1\_P2.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_1\_TOTPROD.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_1\_DIV.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_1\_DPT.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_1\_SUBDPT.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_1\_CLASS.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_1\_SUBCLASS.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_1\_STYLE.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_1\_STYLECOLOR.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_2\_P1.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_2\_TOTORG.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_2\_BU.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_2\_CHANNEL.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_2\_REGION.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_2\_DISTRICT.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_2\_TOUCHPOINTDAT.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_4\_TOTYEAR.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_4\_YEAR.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_4\_SEASON.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_4\_QUARTER.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_4\_PERIOD.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_4\_WEEK.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_AG\_4\_DAY.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_EXT\_EPSD\_ERRORS.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_EXT\_EPSD\_IMPORT\_ATTR\_DEF\_POT.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_EXT\_EPSD\_IMPORT\_MEMBER\_REL.REJ|*[record count]* |

***NOTE****: The control file will only list error files when errors have occurred within the associated data file. If there were no errors in any data files, the .CON file will not be written.*

**Foundation Data Update**

When errors occur during Foundation Data Update processing, an error file is written for each fact data file containing records that failed EP validation. The error files are listed within a control file where each record contains the name of the error file and its record count separated by a **pipe (‘|’) character.**

These files are written to the TY\_Load\_Errors directory.

|  |  |
| --- | --- |
| **CONTROL FILE NAME** | **RECORD FORMAT** |
| *<YYYYMMDDHHMMSS>*\_FACT.CON | *<YYYYMMDDHHMMSS>*\_PKB\_ACTUAL\_INVBAL.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_PKB\_ACTUAL\_MKDNS.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_PKB\_ACTUAL\_ON\_ORDER.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_PKB\_ACTUAL\_OTHINV.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_PKB\_ACTUAL\_RECEIPTS.REJ|*[record count]* |
| *<YYYYMMDDHHMMSS>*\_PKB\_ACTUAL\_SALES.REJ|*[record count]* |

***NOTE****: The control file will only list error files when errors have occurred within the associated data file. If there were no errors in any data files, the .CON file will not be written.*

# Appendix A

### Supporting Documents

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Version** | **Date** |
| EP PlanNOW! Technical Design Document 1.0.pdf | 1.0 | 03-23-2022 |
| EP PlanNOW! Batch Design Document 1.0.docx | 1.0 | 03-23-2022 |
| EP Batch Facility v2020.1.0.0.docx | 2020.1.0.0 | 05-04-2020 |
| Control-M configuration v2020.1.0.0.xlsx | 2020.1.0.0 | 03-03-2022 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |