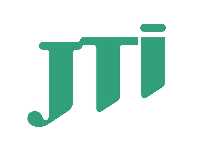
****

**HLFD Interface Contract New In OLTP Version 6.0**

Functional Specifications

Values%20worldmap2-rgb

**Target Release: Feb 2022**

**Sponsor Market: CT**

Document History

**Reviews**

| Version | Name | Date | Signature |
| --- | --- | --- | --- |
| 6.0 | TME - HLFD Interface Contract New In OLTP Version 6.0.docx | 19/11/2021 |  |
|  |  |  |  |

**Approvals**

| Version | Name | Date | Signature |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Revision History

| Version | Section(s) | Summary of Changes | Written By | CR/INC# | Date | PRD Release |
| --- | --- | --- | --- | --- | --- | --- |
| 6.0 |  | Based on Version 5.3 | Jhanvi Jeetun | CHG213427 | 19/11/2021 | Feb 2022 |
|  |  |  |  |  |  |  |

Table of Contents

[1. Introduction 5](#_Toc472442760)

[1.1. Overview 5](#_Toc472442761)

[1.2. Inserts/Update/Delete 6](#_Toc472442762)

[1.3. Systems 6](#_Toc472442763)

[1.4. File Naming Convention 6](#_Toc472442764)

[1.5. File Delimiter 7](#_Toc472442765)

[1.6. Dependencies on other interfaces. 7](#_Toc472442766)

[2. Description of Tables and Fields 8](#_Toc472442767)

[2.1. DB Tables 8](#_Toc472442768)

[2.2. Primary Keys 8](#_Toc472442769)

[2.3. Flat File Fields 9](#_Toc472442770)

[2.4. Fields calculated by the interface 16](#_Toc472442771)

[2.5. Validations 19](#_Toc472442772)

[2.5.1. LOV Checks 19](#_Toc472442773)

[2.5.2. User Key 20](#_Toc472442774)

[2.5.3. Foreign Keys 20](#_Toc472442775)

[2.5.3.1. Agreement Foreign Key Check 21](#_Toc472442776)

[2.5.4. Other Validations 23](#_Toc472442777)

[Managing primary contract account: 23](#_Toc472442778)

[Managing child contracts 23](#_Toc472442779)

[ACTIVE\_FLG and Status 23](#_Toc472442780)

[Cancellation Reason & Cancellation Detail 23](#_Toc472442781)

[Validation on Contracted Start Date 24](#_Toc472442782)

[3. Additional Functionality 25](#_Toc472442783)

[3.1. Fixed template flag processing 25](#_Toc472442784)

[3.2. Additional Notes on field AGR\_X\_FIXEDTMPLFLG 25](#_Toc472442785)

[4. IFB Files 26](#_Toc472442786)

[5. Error Messages 27](#_Toc472442787)

[6. Example of flat file 28](#_Toc472442788)

[7. Interface execution 29](#_Toc472442789)

# Introduction

This document gives the high-level description of the inbound to OLTP TME Interface loader Contract new Version 6.

## Overview

This document describes all fields that can be loaded in the Contract New entity and all fields that are automatically generated during an EIM-load.

Each contract can be assigned to several accounts and each account can have several contracts.

What makes a contract unique is its contract number.

Each contract is linked towards the position of the person that has created the contract – not the creator itself.

The relationship of Contracts (Agreement) can be summarised by the Diagram below:



Fig 1 – Contract Entity relationship

## Inserts/Update/Delete

* *Insert Functionality: ENABLED*

New Contracts will be created via this interface execution when there is no existing record for the values provided on PK columns.

* *Update Functionality: ENABLED*

Existing contracts can be updated via this interface by providing the accurate PK columns values.

Columns provided empty won’t be taken into consideration and only not empty columns values will be used to update the corresponding fields on TME Sales for the contract.

For updating contract columns are not mandatory.

* *Delete Functionality*: No deletion can be performed via this interface.

## Systems

Example:

Only one file per market is allowed for each load for the same system Id and entity version.

The #records will be controlled via the Flat File Load control mechanism.

## File Naming Convention

The file name is formed by concatenating the following fields:

* Country code, 2 characters:
  + IT for Italy
  + ES for Spain
* File Type, 2 characters: varies for each file type
* Version, 2 digits: revision version of the file type
* **System Id**, 2 digits: unique identifier for data source, given by the markets
* An underscore (‘\_’)
* Extraction date, in format YYYYMMDD. If the extraction is done overnight, the past day’s date will be used (e.g., if data extracted between Monday and Tuesday, Monday’s date will be used)
* An underscore (‘\_’)
* A group number. The group ‘10’ will be used for normal extraction. Other groups will be used in exceptional cases (e.g., for data correction).
* File extension: ‘.txt’

Example:

RUCH0602\_20150128\_10.txt

This is a file from Russia, of type CH, version 06, coming from system 02, extracted on January 28th, 2015, and belonging to group 10 (normal group).

## File Delimiter

The field separator is a pipe ‘|’.

## Dependencies on other interfaces.

The corresponding AC (Account), BD (Budget) flat files would have to be loaded prior to loaded the CH. Also the Positions would have to be set up in advance.

Otherwise all records will be rejected.

# Description of Tables and Fields

## DB Tables

OLTP:

* EIM ones: EIM\_AGREEMENT and EIM\_ACCNT\_UT
* Base ones: S\_DOC\_AGREE, S\_ACCNT\_AGR\_CUT, S\_MDF and S\_SRC\_PAYMENT.

## Primary Keys

*Flat File PK:*

The primary keys for this file are:

* AGREE\_IDENTIFIER: Agreement Unique Identifier
* TARGET\_ACCNT\_BU: Target Account Business Unit
* TARGET\_ACCNT\_NAME: Target Account Name
* AGR\_BU: Agreement Business Unit
* AGR\_REV\_NUM: Agreement Revision Number

*Base tables PK:*

The primary keys for S\_DOC\_AGREE are:

* NAME: Agreement Name.
* TARGET\_OU\_ID: Target Account Id.
* REV\_NUM: Revision Num.
* BU\_ID: Agreement Business Unit

The primary keys for S\_ACCNT\_AGR\_CUT are:

* PARTY\_UID: Party Unique Identifier.
* PARTY\_TYPE\_CD: Party Type Code.
* ACCNT\_ID: Account Id.

The primary keys for S\_SRC\_PAYMENT are:

* PAYMENT\_REQ\_NUM: Payment Requirement Number.
* TYPE\_CD: Payment Type.

The primary keys for S\_MDF are:

* BUDGET\_UID: Marketing Development Funds (MDF).
* BUDGET\_BU: Budget Business Unit (Same as AGR\_BU)

## Flat File Fields

| **Order** | **Column Name** | **Type** | **Mandatory/PK** | **Contents** | **Destination Type** | **Clarifications**  **/Format** | **Destination Field location in TME Sales/OBIEE** | **Corresponding EIM TABLE** | **Corresponding BASE TABLE** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | X\_AGR\_NAME | C100 | Y | Name of Agreement |  |  | Name | EIM\_AGREEMENT.X\_NAME | S\_DOC\_AGREE.X\_NAME |
| 2 | TARGET\_ACCNT\_NAME | C100 | Y (PK) | Code of Account to which the contract is linked | FK to S\_ORG\_EXT  FK to S\_DOC\_AGREE | (refer section 2.5.3.1) | Account Name | EIM\_AGREEMENT.  TARGET\_OU\_ID | S\_DOC\_AGREE. TARGET\_ACCNT\_NAME |
| 3 | AGR\_REV\_NUM | N22,7 | Y (PK) | Revision Number. It should always be 1 | FK to S\_DOC\_AGREE | (refer section 2.5.3.1) | n/a | EIM\_AGREEMENT. AGR\_REV\_NUM | S\_DOC\_AGREE. REV\_NUM |
| 4 | AGR\_BU | C100 | Y (PK) | Business unit for agreement, always refers to a default business unit e.g "Spain, ES" or “Italy, IT” | FK to S\_DOC\_AGREE | (refer section 2.5.3.1) | n/a | EIM\_AGREEMENT.AGR\_BU | S\_DOC\_AGREE.BU\_ID |
| 5 | AGR\_AGREE\_NUM | C30 | Y (no PK) | Agreement Number |  |  | Agreement Number | EIM\_AGREEMENT.AGR\_AGREE\_NUM | S\_DOC\_AGREE.AGREE\_NUM |
| 6 | AGR\_ACTL\_COST\_AMT | N22,7 | N | Actual Cost Amount |  | For Outlet, Chain or KA Agreements, it signifies the Fixed Amount on the contract.  This value cannot be null in Siebel Sales. So that if in the flat file the value is null:  - In case of new contracts a value 0 is entered  - In case of existing contracts, existing value remains. | Actual Cost Amount | EIM\_AGREEMENT.AGR\_ACTL\_COST\_AMT | S\_DOC\_AGREE.ACTL\_COST\_AMT |
| 7 | AGR\_BL\_CURCY\_CD | C20 | N | Billing Currency Code |  |  | n/a | EIM\_AGREEMENT.AGR\_BL\_CURCY\_CD | S\_DOC\_AGREE.BL\_CURCY\_CD |
| 8 | AGR\_DESC\_TEXT | C1000 | N | Description |  |  | Full Description | EIM\_AGREEMENT.X\_DESC\_TEXT | S\_DOC\_AGREE.X\_DESC\_TEXT |
| 9 | AGR\_EFF\_END\_DT | D | N | Effective End Date |  |  | End Date | EIM\_AGREEMENT.AGR\_EFF\_END\_DT | S\_DOC\_AGREE.EFF\_END\_DT |
| 10 | AGR\_CNTRSRVTYPE\_CD | C30 | Y | Level of Agreement | MLOV:SRV\_AGREE\_TYPE | The chosen value will determine how the contract is shown as in the sales application different views are available Outlet, Chain or KA Agreements:  - For Outlet Agreements, the value will be 'Outlet Agreement'  - For Chain Agreements, the value must be 'Chain'  - For KA Agreements, the value must be 'Key Account' | Level of Agreement | EIM\_AGREEMENT. AGR\_CNTRSRVTYPE\_CD | S\_DOC\_AGREE.CNTR\_SRV\_TYPE\_CD |
| 11 | AGR\_EFF\_START\_DT | D | Y | Start Date |  | YYYY/MM/DD  Start Date <= End Date | Start Date | EIM\_AGREEMENT.AGR\_EFF\_START\_DT | S\_DOC\_AGREE. EFF\_START\_DT |
| 12 | AGR\_STAT\_CD | C30 | N | Status Code (Mandatory only for new Contract) | MLOV: SRV\_AGREE\_STATUS |  | Status | EIM\_AGREEMENT.AGR\_STAT\_CD | S\_DOC\_AGREE.STAT\_CD |
| 13 | AGR\_AGREE\_CD | C30 | N | Agreement Type Code (Mandatory only for new Contract) | MLOV: AGREEMENT\_TYPE |  | Type | EIM\_AGREEMENT. AGR\_AGREE\_CD | S\_DOC\_AGREE. AGREE\_CD |
| 14 | PSTN\_POSTN\_NAME | C50 | Y | Position name for Contract. Position of creator of contract | FK to S\_POSTN |  | n/a | EIM\_AGREEMENT. PSTN\_POSTN\_NAME | S\_DOC\_AGREE.POSTN\_ID |
| 15 | X\_PAYMENT\_TERMS | C30 | N | Payment terms | MLOV: JTI\_PAYMENT\_TERMS |  | Payment Terms | EIM\_AGREEMENT.X\_PAYMENT\_TERMS | S\_DOC\_AGREE.X\_PAYMENT\_TERMS |
| 16 | AGR\_PO\_NUM | C30 | N | Purchase Order Number |  |  | PO # | EIM\_AGREEMENT. AGR\_PO\_NUM | S\_DOC\_AGREE.PO\_NUM |
| 17 | X\_SALES\_ORG\_CD | C30 | N | Vendor | MLOV: SALES\_ORGANIZATION |  | Vendor | EIM\_AGREEMENT.X\_SALES\_ORG\_CD | S\_DOC\_AGREE.X\_SALES\_ORG\_CD |
| 18 | X\_PRIMARY\_ACCNT\_FLG | C1 | Y | Account Primary Flag |  | [KA Contracts]  Only one record per ~~AGR\_AGREE\_NUM~~ AGREE\_IDENTIFIER will have this field set to Y. if not, the record will be rejected.  In case of KA Contracts, only the primary account will have this flag as Y and the rest of them as N.  All the agreement details will remain same for all the accounts in the KA Contract  [Outlet agreements]  The flag must be Y and only one account is linked to one agreement.  [Other Contracts]  For agreements with multiple accounts this flag indicates the primary account.  When “Y” all other contract attributes can be provided. When “N” only the fields X\_AGR\_NAME, TARGET\_ACCNT\_BU, TARGET\_ACCNT\_NAME, AGR\_REV\_NUM, AGR\_BU, VIS\_BU, AGR\_AGREE\_NUM,  AGREE\_IDENTIFIER  X\_AGR\_EST\_COST\_AMT X\_PRIMARY\_ACCNT\_FLG & X\_ACCOUNT\_PERCENTAGE will be processed. | Primary Account | n/a | n/a |
| 19 | AGRXPAYMENTMETHODC | C30 | N | Payment Method | MLOV: JTI\_PAYMENT\_METHOD |  | n/a | EIM\_AGREEMENT.X\_PAYMENT\_METHOD\_CD | S\_DOC\_AGREE.AGRXPAYMENTMETHODC |
| 20 | AGR\_X\_FIXEDTMPLFLG | C1 | N | Fixed Template Flag |  | Fixed Template Flag.  Flag indicating if the budget, agreement type, payment plan template, amount, contract terms, awards products, incentives can be changed or not.  Null or N: can be changed,  Y: cannot.  Field is only available for update for contract templates. **When set to “Y” contract items for the contract cannot be loaded.**  [KA Contracts].  Is not applicable to KA Contracts. The value must be N or Null. **See more info on additional notes section.** | Fixed Template | EIM\_AGREEMENT.X\_FIXED\_TMPL\_FLG | S\_DOC\_AGREE.AGR\_X\_FIXEDTMPLFLG |
| 21 | BUDGET\_UID | C30 | N | Marketing Development Funds (MDF) Unique Identifier | FK to S\_MDF |  | n/a | EIM\_AGREEMENT.AGR\_X\_MDF\_UID | S\_DOC\_AGREE.X\_BUDGET\_ID |
| 22 | BUDGET\_NAME | C50 | N | Marketing Development Funds (MDF) Name | FK to S\_MDF |  | Budget | n/a | n/a |
| 23 | AGR\_ACTIVE\_FLG | C1 | N | Active Flag |  | Sets the contract active Flag e.g. “ Y” or “N”  The value must be Y only if the status is set as “Active”. Else “N”.  If provided with NULL:  - In case of new contracts it will be set:  - to “Y” if status is Active  - to “N” for any other status.  - In case of existing contracts, existing value remains.  ( see Section | n/a | EIM\_AGREEMENT.AGR\_ACTIVE\_FLG | S\_DOC\_AGREE.ACTIVE\_FLG |
| 24 | X\_CONTACT\_GROUP\_CD | C30 | N | Contact group | MLOV: JTI\_CONTACT\_GROUP |  | JTI Contract Group | EIM\_AGREEMENT.X\_CONTACT\_GROUP\_CD | S\_DOC\_AGREE.X\_CONTACT\_GROUP\_CD |
| 25 | X\_PR\_TEXT | C30 | N | JTI PR Number |  |  | SAP PR# | EIM\_AGREEMENT. X\_PR\_TEXT | S\_DOC\_AGREE. X\_PR\_TEXT |
| 26 | X\_VENDOR\_TEXT | C7 | N | JTI Vendor |  |  | SAP Vendor# | EIM\_AGREEMENT.X\_VENDOR\_TEXT | S\_DOC\_AGREE. X\_VENDOR\_TEXT |
| 27 | X\_MAX\_TXN\_PTS\_NUM | N22,7 | N | Max Points per Trans |  | For Awards Products Loyalty program  Set only if RLP Flag is Y | Max # points by transaction | EIM\_AGREEMENT.X\_MAX\_TXN\_PTS\_NUM | S\_DOC\_AGREE.X\_MAX\_TXN\_PTS\_NUM |
| 28 | X\_MAX\_PTS\_NUM | N22,7 | N | Max Points |  | For Awards Products Loyalty program  Set only if RLP Flag is Y | Max # Points | EIM\_AGREEMENT.X\_MAX\_PTS\_NUM | S\_DOC\_AGREE.X\_MAX\_PTS\_NUM |
| 29 | X\_RLP\_PTS\_VAL\_NUM | N22,7 | N | Points Value |  | For Awards Products Loyalty program  Set only if RLP Flag is Y | Points Value | EIM\_AGREEMENT. X\_RLP\_PTS\_VAL\_NUM | S\_DOC\_AGREE. X\_RLP\_PTS\_VAL\_NUM |
| 30 | X\_RLP\_FLG | C1 | N | RLP Flag |  |  | Loyalty Program | EIM\_AGREEMENT. X\_RLP\_FLG | S\_DOC\_AGREE. X\_RLP\_FLG |
| 31 | X\_SIGNATURE\_DATE | D | N | Signing Date |  | YYYY/MM/DD | Signature Date | EIM\_AGREEMENT.X\_SIGNATURE\_DATE | S\_DOC\_AGREE.X\_SIGNATURE\_DATE |
| 32 | AVG\_AMT\_PER\_OUTLET | N22,7 | N | Average Amt. per Outlet |  | Applicable only to KA and Chain Agreements. This is the amount assigned to each outlet.  This value cannot be null in Siebel Sales. So that if in the flat file the value is null:  - In case of new contracts a value 0 is entered  - In case of existing contracts, existing value remains.  For Outlet Agreement the value is always 0 | Avg. Amount per Outlet | EIM\_AGREEMENT. X\_AVG\_OUTLET\_AMT | S\_DOC\_AGREE. X\_AVG\_OUTLET\_AMT |
| 33 | CONFIDENTIAL\_FLG | C1 | N | Confidential flag |  | Applicable only to KA Agreements. Can have Y or N values.  This value cannot be null in Siebel Sales. So that if in the flat file the value is null:  - In case of new contracts “N” value is entered  - In case of existing contracts, existing value remains  For Outlet Agreement, it should always be N | Confidential | EIM\_AGREEMENT.AGRXCONFIDENTIALFL | S\_DOC\_AGREE.X\_CONFIDENTIAL\_FLG |
| 34 | ACT\_END\_DATE | D | N | Actual End Date |  | YYYY/MM/DD | Closing Date | EIM\_AGREEMENT.X\_ACTUAL\_END\_DT | S\_DOC\_AGREE.X\_ACTUAL\_END\_DT |
| 35 | READ\_ONLY\_FLG | C1 | N | Read Only Flag |  | If the value is Y, no updates/inserts can be performed on the agreement header and its child entities | Read Only flag | EIM\_AGREEMENT. X\_READ\_ONLY\_FLG | S\_DOC\_AGREE. X\_READ\_ONLY\_FLG |
| 36 | BUSINESS\_REASON | C30 | N | Business Reason: | MLOV:JTI\_BUSINESS\_REASON |  | Business Reason | EIM\_AGREEMENT. X\_BUSINESS\_REASON\_CD | S\_DOC\_AGREE.X\_BUSINESS\_REASON\_CD |
| 37 | X\_NO\_OF\_YEARS | N22,7 | N | Number of years of the contract. |  | This value cannot be null in Siebel Sales. So that if in the flat file the value is null:  - In case of new contracts '1' value is entered  - In case of existing contracts, existing value remains. | No. of Years | EIM\_AGREEMENT.X\_NO\_YEARS\_NUM | S\_DOC\_AGREE.X\_NO\_YEARS\_NUM |
| 38 | X\_TXT\_ATTRIBUTE1 | C30 | N | Text Attribute 1 |  |  | Text Attribute1 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_01 | S\_DOC\_AGREE. X\_ATTRIB\_01 |
| 39 | X\_TXT\_ATTRIBUTE2 | C250 | N | Text Attribute 2 |  |  | Text Attribute2 | EIM\_AGREEMENT.AGR\_X\_ATTRIB\_02 | S\_DOC\_AGREE. X\_ATTRIB\_02 |
| 40 | X\_TXT\_ATTRIBUTE3 | C250 | N | Text Attribute 3 |  |  | Text Attribute3 | EIM\_AGREEMENT.AGR\_X\_ATTRIB\_03 | S\_DOC\_AGREE.X\_ATTRIB\_03 |
| 41 | X\_TXT\_ATTRIBUTE4 | C250 | N | Text Attribute 4 |  |  | Text Attribute4 | EIM\_AGREEMENT.AGR\_X\_ATTRIB\_04 | S\_DOC\_AGREE.X\_ATTRIB\_04 |
| 42 | X\_TXT\_ATTRIBUTE5 | C30 | N | Text Attribute 5 |  |  | Text Attribute5 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_05 | S\_DOC\_AGREE. X\_ATTRIB\_05 |
| 43 | X\_TXT\_ATTRIBUTE6 | C30 | N | Text Attribute 6 |  |  | Text Attribute6 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_06 | S\_DOC\_AGREE. X\_ATTRIB\_06 |
| 44 | X\_TXT\_ATTRIBUTE7 | C30 | N | Text Attribute 7 |  |  | Text Attribute7 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_07 | S\_DOC\_AGREE. X\_ATTRIB\_07 |
| 45 | X\_TXT\_ATTRIBUTE8 | C30 | N | Text Attribute 8 |  |  | Text Attribute8 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_08 | S\_DOC\_AGREE. X\_ATTRIB\_08 |
| 46 | X\_TXT\_ATTRIBUTE9 | C30 | N | Text Attribute 9 |  |  | Text Attribute9 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_09 | S\_DOC\_AGREE. X\_ATTRIB\_09 |
| 47 | X\_TXT\_ATTRIBUTE10 | C30 | N | Text Attribute 10 |  |  | Text Attribute10 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_10 | S\_DOC\_AGREE. X\_ATTRIB\_10 |
| 48 | X\_TXT\_ATTRIBUTE11 | C250 | N | Text Attribute 11 |  |  | Text Attribute11 | EIM\_AGREEMENT.AGR\_X\_ATTRIB\_11 | S\_DOC\_AGREE.X\_ATTRIB\_11 |
| 49 | X\_TXT\_ATTRIBUTE12 | C30 | N | Text Attribute 12 |  |  | Text Attribute12 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_12 | S\_DOC\_AGREE. X\_ATTRIB\_12 |
| 50 | X\_LOV\_ATTRIBUTE1 | C30 | N | LOV Attribute 1 | MLOV:JTI\_AGR\_ATTRIB\_13 |  | LOV Attribute1 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_13 | S\_DOC\_AGREE. X\_ATTRIB\_13 |
| 51 | X\_LOV\_ATTRIBUTE2 | C30 | N | LOV Attribute 2 | MLOV:JTI\_AGR\_ATTRIB\_14 |  | LOV Attribute2 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_14 | S\_DOC\_AGREE. X\_ATTRIB\_14 |
| 52 | X\_LOV\_ATTRIBUTE3 | C30 | N | LOV Attribute 3 | MLOV:JTI\_AGR\_ATTRIB\_15 |  | LOV Attribute3 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_15 | S\_DOC\_AGREE. X\_ATTRIB\_15 |
| 53 | X\_LOV\_ATTRIBUTE4 | C30 | N | LOV Attribute 4 | MLOV:JTI\_AGR\_ATTRIB\_16 |  | LOV Attribute4 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_16 | S\_DOC\_AGREE. X\_ATTRIB\_16 |
| 54 | X\_LOV\_ATTRIBUTE5 | C30 | N | LOV Attribute 5 | MLOV:JTI\_AGR\_ATTRIB\_17 |  | LOV Attribute5 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_17 | S\_DOC\_AGREE. X\_ATTRIB\_17 |
| 55 | X\_NUM\_ATTRIBUTE1 | N22,7 | N | Numeric Attribute 1 |  |  | Num Attribute1 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_18 | S\_DOC\_AGREE. X\_ATTRIB\_18 |
| 56 | X\_NUM\_ATTRIBUTE2 | N22,7 | N | Numeric Attribute 2 |  |  | Num Attribute2 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_19 | S\_DOC\_AGREE. X\_ATTRIB\_19 |
| 57 | X\_NUM\_ATTRIBUTE3 | N22,7 | N | Numeric Attribute 3 |  |  | Num Attribute3 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_20 | S\_DOC\_AGREE. X\_ATTRIB\_20 |
| 58 | X\_NUM\_ATTRIBUTE4 | N22,7 | N | Numeric Attribute4 |  |  | Num Attribute4 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_21 | S\_DOC\_AGREE. X\_ATTRIB\_21 |
| 59 | X\_NUM\_ATTRIBUTE5 | N22,7 | N | Numeric Attribute 5 |  |  | Num Attribute5 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_22 | S\_DOC\_AGREE. X\_ATTRIB\_22 |
| 60 | X\_ADMIN\_LOV\_ATTRIBUTE1 | C30 | N | Local Administrator LOV Attribute 1 | MLOV: JTI\_AGR\_ATTRIB\_23 |  | LA LOV Attribute1 | EIM\_AGREEMENT. AGR\_X\_ATTRIB\_23 | S\_DOC\_AGREE. X\_ATTRIB\_23 |
| 61 | X\_RESPONSIBLE\_PERSON | C50 | N | Responsible Person of the contract | FK to S\_USER |  | Responsible Person | EIM\_AGREEMENT. AGR\_X\_RESP\_LOGIN  EIM\_AGREEMENT. AGR\_XRESPLOGINDOMA  EIM\_AGREEMENT. AGR\_X\_RESP\_PERTEXT | S\_DOC\_AGREE. X\_RESP\_PER\_ID  S\_DOC\_AGREE. X\_RESP\_PER\_TEXT |
| 62 | X\_OBJECTIVE\_ID | C30 | N | Objective id |  |  |  | EIM\_AGREE\_LS.AGRXM\_ATTRIB\_04 | S\_DOC\_AGREE\_XM. ATTRIB\_04 |
| 63 | X\_CANCELLATION\_REASON\_CD | C30 | N | Cancellation Reason | MLOV: JTI\_CONTRACT\_CANCEL\_REASON |  | Cancellation Reason | EIM\_AGREEMENT. X\_CONTR\_CANCEL\_REASON\_CD | S\_DOC\_AGREE. X\_CONTR\_CANCEL\_REASON\_CD |
| 64 | X\_CANCELLATION\_DETAILS\_CD | C30 | N | Cancellation Details | MLOV: JTI\_CONTRACT\_CANCEL\_DETAIL |  | Cancellation Details | EIM\_AGREEMENT. X\_CONTR\_CANCEL\_DTL\_CD | S\_DOC\_AGREE. X\_CONTR\_CANCEL\_DTL\_CD |
| 65 | X\_CANCELLATION\_CLARIFICATIONS\_CD | C30 | N | Details clarifications | MLOV: JTI\_CONTRACT\_CANCEL\_DTL\_CLAR |  | Details clarifications | EIM\_AGREEMENT. X\_CONTR\_CANCEL\_DTL\_CLF\_CD | S\_DOC\_AGREE. X\_CONTR\_CANCEL\_DTL\_CLF\_CD |
| 66 | X\_CANCELLATION\_DATE | C30 | N | Details clarifications |  | Field is mandatory when status is Cancelled only. Should be Null when status other than 'Cancelled'. | Cancellation Date | EIM\_AGREEMENT. X\_CONTR\_CANCEL\_DT | S\_DOC\_AGREE. X\_CONTR\_CANCEL\_DT |
| 67 | ORIGINAL\_AGR\_NUM | C30 | N | Agreement Num of Original Agreement | FK to S\_DOC\_AGREE  (refer section 2.5.3.1) | If new contract is created and Original Contract is null then Original Contract Id = Contract Id | Original Contract Id | EIM\_AGREEMENT. AGR\_XORIGINALAGREE | S\_DOC\_AGREE. X\_ORIGINAL\_CONTRACT\_ID |
| 68 | ORIGINAL\_TARGET\_ACCNT | C100 | N | Account Code to which the original contract is linked | FK to S\_DOC\_AGREE |  | Original Contract Id | ~~EIM\_AGREEMENT. AGRXORIGINALAGREE2~~ | ~~S\_DOC\_AGREE. X\_ORIGINAL\_CONTRACT\_ID~~ |
| 69 | AGR\_PROD\_CODE | C100 | N | Brand | FK to S\_PROD\_INT |  | Product Code | EIM\_AGREEMENT.AGR\_PROD\_NAME | S\_DOC\_AGREE.PRODUCT\_ID |
| 70 | AGREE\_IDENTIFIER | C50 | Y(PK) | Agreement Unique Identifier | FK to S\_DOC\_AGREE | (refer section 2.5.3.1) | Agreement Unique Identifier | EIM\_AGREEMENT.AGR\_NAME | S\_DOC\_AGREE.NAME |
| 71 | CURCY\_DT | D |  | Exchange Date |  | YYYY/MM/DD | Exchange Date | EIM\_AGREEMENT.AGR\_AGREE\_EXCH\_DT | S\_DOC\_AGREE.AGREE\_EXCH\_DT |
| 72 | ORIGINAL\_AGR\_IDENTIFIER | C50 | N | Agreement Unique Identifier of Original Agreement | FK to S\_DOC\_AGREE  (refer section 2.5.3.1) | If new contract is created and Original Contract is null then Original Contract Id = Contract Id | Original Contract Id | EIM\_AGREEMENT. AGR\_XORIGINALAGREE | S\_DOC\_AGREE.X\_ORIGINAL\_CONTRACT\_ID |
| 73 | LOV\_ATTRIBUTE\_6 | C30 | N | LOV Attribute 6 | MLOV: JTI\_ATTRIB\_24 |  | LOV Attribute 6 | EIM\_AGREEMENT.X\_ATTRIB\_24 | S\_DOC\_AGREE.X\_ATTRIB\_24 |
| 74 | LOV\_ATTRIBUTE\_7 | C30 | N | LOV Attribute 7 | MLOV: JTI\_ATTRIB\_25 |  | LOV Attribute 7 | EIM\_AGREEMENT.X\_ATTRIB\_25 | S\_DOC\_AGREE.X\_ATTRIB\_25 |
| 75 | PAR\_AGREE\_IDENTIFIER | C50 | N | Parent Agreement Unique Identifier | FK to S\_DOC\_AGREE | The Parent Agreement should already exists on Siebel  (refer section 2.5.3.1) | Parent Agreement | EIM\_AGREEMENT.PAR\_AGREE\_NAME | S\_DOC\_AGREE.PAR\_AGREE\_ID |
| 76 | TEMPLATE\_IDENTIFIER | C50 | N | Template Agreement Unique Identifier | FK to S\_DOC\_AGREE | (refer section 2.5.3.1) | Template Agreement | EIM\_AGREEMENT.AGR\_X\_TEMPL\_ID | S\_DOC\_AGREE.X\_TEMPL\_ID |
| 77 | X\_AGR\_CATEGORY\_CD | C30 | N | Category | MLOV: JTI\_AGR\_CATEGORY |  | Category | EIM\_AGREEMENT.X\_AGR\_CATEGORY\_CD | S\_DOC\_AGREE.X\_AGR\_CATEGORY\_CD |
| 78 | X\_AGR\_HIERARCHY\_TYPE\_CD | C30 | N | Hierarchy Type | MLOV: JTI\_AGR\_HIERARCHY\_TYPE |  | Hierarchy Type | EIM\_AGREEMENT.X\_AGR\_HIERARCHY\_TYPE\_CD | S\_DOC\_AGREE.X\_AGR\_HIERARCHY\_TYPE\_CD |
| 79 | X\_CONTRACTED\_START\_DT | D | N | Contracted Start Date |  | YYYY/MM/DD |  | EIM\_ACCNT\_UT | S\_ACCNT\_AGR\_CUT. X\_CONTRACTED\_START\_DT |
| 80 | X\_CONTRACTED\_END\_DT | D | N | Contracted End Date |  | YYYY/MM/DD |  | EIM\_ACCNT\_UT | S\_ACCNT\_AGR\_CUT. X\_CONTRACTED\_END\_DT |
| 81 | X\_TOTAL\_AMOUNT\_LIMIT | N22,7 | N | Total Amount Limit |  |  | JTI Total Amount Limit | EIM\_AGREEMENT. X\_TOTAL\_AMT\_LIMIT\_NUM | S\_DOC\_AGREE. X\_TOTAL\_AMT\_LIMIT\_NUM |
| 82 | X\_TOTAL\_POINTS\_LIMIT | N22,7 | N | Total Points Limit |  |  | JTI Total Points Limit | EIM\_AGREEMENT. X\_TOTAL\_PTS\_LIMIT\_NUM | S\_DOC\_AGREE. X\_TOTAL\_PTS\_LIMIT\_NUM |

## Fields calculated by the interface

When X\_PRIMARY\_ACCNT\_FLG is “Y”, a contract header will be created and all fields provided will be taken into account. The account provided will be set as the Primary Account of the contract. On top of that, the account should be linked to the contract in the table S\_ACCNT\_AGR\_CUT.

When X\_PRIMARY\_ACCNT\_FLG is “N”, the accounts needs to be added as secondary to the contract by inserting the record in the table S\_ACCNT\_AGR\_CUT. Only the fields X\_AGR\_NAME, TARGET\_ACCNT\_BU, TARGET\_ACCNT\_NAME, AGR\_REV\_NUM, AGR\_BU, VIS\_BU, AGR\_AGREE\_NUM, AGREE\_IDENTIFIER, X\_PRIMARY\_ACCNT\_FLG, X\_AGR\_EST\_COST\_AMT & X\_ACCOUNT\_PERCENTAGE will be processed if the flag is set to ‘N’.

The file is the main source of data for table S\_DOC\_AGREE

The loader will process data provided in the flat file in 2 steps:

1. BATCH 1 : Will contain all records where X\_ACCNT\_PRIMARY\_FLG = “Y”. This batch will create the contract header :
   1. Create a new record in S\_DOC\_AGREE based on user key fields
   2. All fields of the records (foreign keys/attributes) will be processed
2. BATCH 2 : Will contain all records where X\_ACCNT\_PRIMARY\_FLG = “N” and “Y”. This batch will simply add secondary account links to the existing/newly created contract header :
   1. Create a new record in S\_ACCNT\_AGR\_CUT to represent the new Contract secondary link to account
   2. Only value X\_AGR\_EST\_COST\_AMT & X\_ACCOUNT\_PERCENTAGE will be processed
   3. All other fields besides the above value and contract/account users keys are ignored

The loader must also create 2 new records per header contract in the table S\_SRC\_PAYMENT

* Create a Check book in the S\_SRC\_PAYMENT where PAY\_TYPE\_CD will be “Check”
* Create a payment in S\_SRC\_PAYMENT where PAY\_TYPE\_CD will be “Deduction”

The generation of payment and Check book is parameterised in table WC\_ETL\_MARKET\_PARAM\_G where ETL\_PROCESS\_NAME= "Generate Cheque and Payment". To activate a market, the column EXECUTION\_FLG should be set to ‘Y’ for the corresponding market.

AGR\_ACTIVE\_FLG field value calculated during EIM Process:

The value must be Y only if the status is set as 'Active'. Else 'N'.

If provided with NULL:

In case of new contracts it will be set:

* to 'Y' if status is Active
* to 'N' for any other status

In case of existing contracts, existing value remains.

The following fields are not loaded from incoming files, but are calculated during the EIM process:

|  |  |  |
| --- | --- | --- |
| **Description** | **Corresponding column in**  **EIM interface table** | **Corresponding Column in Base table** |
| The value must be Y only if the status is set as 'Active'. Else 'N'.  If provided with NULL:  In case of new contracts it will be set:   * to 'Y' if status is Active * to 'N' for any other status   In case of existing contracts, existing value remains. | AGR\_ACTIVE\_FLG | S\_DOC\_AGREE |
| Always “Y” | AGR\_CONTRACT\_FLG | S\_DOC\_AGREE |
| N | BANK\_DEPOSIT\_FLG | S\_SRC\_PAYMENT |
| N | CG\_PYMT\_SUBMIT\_FLG | S\_SRC\_PAYMENT |
| N | DEPOSIT\_FLG | S\_SRC\_PAYMENT |
| N | DOCNUM\_GNRTD\_FLG | S\_SRC\_PAYMENT |
| N | IC\_TXN\_IMPORT\_FLG | S\_SRC\_PAYMENT |
| Lookup the S\_DOC\_AGREE.ROW\_ID | PAYMENT\_REQ\_NUM | S\_SRC\_PAYMENT |
| N | PHONE\_AUTH\_FLG | S\_SRC\_PAYMENT |
| N | PREAUTH\_FLG | S\_SRC\_PAYMENT |
| N | SELECT\_FLG | S\_SRC\_PAYMENT |
| Fixed value “Payment”. LOV controlled by PAYMENT\_TYPE\_CODE. Non translatable LOV so must be loaded in ENU | TYPE\_CD | S\_SRC\_PAYMENT |
| Lookup the  S\_DOC\_AGREE.ROW\_ID | CONTRACT\_ID | S\_SRC\_PAYMENT |
| X\_AGR\_EST\_COST\_AMT | X\_AMOUNT | S\_ACCNT\_AGR\_CUT |
| X\_ACCOUNT\_PERCENTAGE | X\_AMT\_PERC | S\_ACCNT\_AGR\_CUT |
| LOV controlled by PAYMENT\_METHOD\_CODE. Can be translated value. The field value will be “Check” for the checkbook and “Deduction” for payment | PAY\_PAY\_TYPE\_CD | S\_SRC\_PAYMENT |
| Lookup the S\_ORG\_EXT.ROW\_ID. | ACCNT\_ID | S\_SRC\_PAYMENT |
| This is the amount assigned to each outlet.  This value cannot be null in Siebel Sales. So that if in the flat file the value is null:  - In case of new contracts a value 0 is entered  - In case of existing contracts, existing value remains. | X\_AVG\_OUTLET\_AMT | S\_DOC\_AGREE |
| Lookup on S\_ORG\_EXT based on PAR\_AGREE\_ACC\_NAME value provided | PAR\_AGREE\_ACC\_BU | S\_DOC\_AGREE.PAR\_AGREE\_ID |
|  | PAR\_AGREE\_ACC\_LOC | S\_DOC\_AGREE.PAR\_AGREE\_ID |
| Lookup on S\_DOC\_AGREE based on PAR\_AGREE\_NAME value provided | PAR\_AGREE\_BU | S\_DOC\_AGREE.PAR\_AGREE\_ID |
|  | PAR\_AGREE\_REVNUM | S\_DOC\_AGREE.PAR\_AGREE\_ID |
| A lookup will be performed on S\_LST\_OF\_VAL table with  INPUT  LOV Type : JTI\_MARKET\_SETTINGS  NAME: Period Name (as provided in flat file)  OUTPUT:  VAL: Internal Order Value | X\_VM\_INTERNAL\_ORDER | S\_DOC\_AGREE. X\_VM\_INTERNAL\_ORDER |
| Lookup on S\_PERIOD based on PERIOD\_NAME provided | AGR\_X\_VM\_PERIOD\_BI, AGR\_X\_VM\_PERIOD\_BU | S\_DOC\_AGREE.X\_VM\_PERIOD\_ID |
| Field EIM\_AGREEMENT.AGR\_XRESPLOGINDOMA will always be set as NULL. | X\_RESPONSIBLE\_PERSON | S\_DOC\_AGREE. X\_RESP\_PER\_ID |
| This value cannot be null in Siebel Sales. So that if in the flat file the valu is null:  - In case of new contracts a value 0 is entered  - In case of existing contracts, existing value remains. | AGR\_ACTL\_COST\_AMT | S\_DOC\_AGREE.ACTL\_COST\_AMT |
| Lookup on S\_PROD\_INT based on AGR\_PROD\_CODE | AGR\_PROD\_BI  AGR\_PROD\_BU  AGR\_PROD\_NAME  AGR\_PROD\_VEN\_BU  AGR\_PROD\_VEN\_BI  AGR\_PROD\_VEN\_LOC  AGR\_PROD\_VEN\_NAME | S\_DOC\_AGREE.PRODUCT\_ID |
| Derived from the field X\_RESP\_PER\_ID | EIM\_AGREEMENT. AGR\_X\_RESP\_PERTEXT | S\_DOC\_AGREE. X\_RESP\_PER\_TEXT |
| Populate Business organization field for payment | EIM\_PAYMENT.X\_PAY\_BI, EIM\_PAYMENT.X\_PAY\_BU | S\_SRC\_PAYMENT.X\_BU\_ID |
| If agreement already exists, defaulted to NULL else if flat file column is NULL then default to Organization Currency | EIM\_AGREEMENT.AGR\_BL\_CURCY\_CD | S\_DOC\_AGREE.BL\_CURCY\_CD |
| If agreement already exists, defaulted to NULL else if flat file column is NULL then default to Today. | EIM\_AGREEMENT.AGR\_AGREE\_EXCH\_DT | S\_DOC\_AGREE.AGREE\_EXCH\_DT |
| For new agreements, field is defaulted to '1' | EIM\_AGREEMENT .AGR\_AMENDMENT | S\_DOC\_AGREE.AMENDMENT |
| Defaulted to ‘Default’ (LOVB: JTI\_MODIF\_REASON) | EIM\_PAYMENT.X\_MODIF\_REASON\_CD | S\_SRC\_PAYMENT.X\_MODIF\_REASON\_CD |

## Validations

## LOV Checks

Verify if the LOV of these TYPE exit and if they are active

|  |  |
| --- | --- |
| **Column Name** | **LOV Type** |
| AGR\_CNTRSRVTYPE\_CD | SRV\_AGREE\_TYPE |
| AGR\_STAT\_CD | SRV\_AGREE\_STATUS |
| AGR\_AGREE\_CD | AGREEMENT\_TYPE |
| X\_PAYMENT\_TERMS | JTI\_PAYMENT\_TERMS |
| X\_SALES\_ORG\_CD | SALES\_ORGANIZATION |
| AGRXPAYMENTMETHODC | JTI\_PAYMENT\_METHOD |
| AGRXCONCALCMETHODC | JTI\_CON\_AMT\_CALC\_METHOD |
| “Payment” | PAYMENT\_TYPE\_CODE |
| “Check” & “Deduction” | PAYMENT\_METHOD\_CODE |
| X\_CONTACT\_GROUP\_CD | JTI\_CONTACT\_GROUP |
| BUSINESS\_REASON | JTI\_BUSINESS\_REASON |
| X\_LOV\_ATTRIBUTE1 | JTI\_AGR\_ATTRIB\_13 |
| X\_LOV\_ATTRIBUTE2 | JTI\_AGR\_ATTRIB\_14 |
| X\_LOV\_ATTRIBUTE3 | JTI\_AGR\_ATTRIB\_15 |
| X\_LOV\_ATTRIBUTE4 | JTI\_AGR\_ATTRIB\_16 |
| X\_LOV\_ATTRIBUTE5 | JTI\_AGR\_ATTRIB\_17 |
| X\_ADMIN\_LOV\_ATTRIBUTE1 | JTI\_AGR\_ATTRIB\_23 |
| X\_CANCELLATION\_REASON\_CD | JTI\_CONTRACT\_CANCEL\_REASON |
| X\_CANCELLATION\_DETAILS\_CD | JTI\_CONTRACT\_CANCEL\_DETAIL |
| X\_CANCELLATION\_CLARIFICATIONS\_CD | JTI\_CONTRACT\_CANCEL\_DTL\_CLAR |
| LOV\_ATTRIBUTE\_6 | JTI\_ATTRIB\_24 |
| LOV\_ATTRIBUTE\_7 | JTI\_ATTRIB\_25 |
| X\_AGR\_CATEGORY\_CD | JTI\_AGR\_CATEGORY |
| X\_AGR\_HIERARCHY\_TYPE\_CD | JTI\_AGR\_HIERARCHY\_TYPE |

## User Key

Verify if there are duplicate in the EIM tables though its user key.

Contract keys:

**EIM\_AGREEMENT🡪S\_DOC\_AGREE**

EIM\_AGREEMENT.AGR\_NAME

EIM\_AGREEMENT.TARGET\_ACCNT\_BU

EIM\_AGREEMENT.TARGET\_ACCNT\_NAME

EIM\_AGREEMENT.AGR\_BU

## Foreign Keys

There should be having a Check on the Foreign Keys to verify if EIM will resolve a specific row:

EIM\_AGREEMENT

**S\_ORG\_EXT (Account)**

TARGET\_ACCNT\_BU🡪S\_BU.ROW\_ID

TARGET\_ACCNT\_NAME🡪S\_ORG\_EXT.NAME

**S\_MDF (Budgets)**

BUDGET\_UID🡪S\_MDF.ROW\_ID

BUDGET\_NAME🡪S\_MDF.NAME

BUDGET\_BU🡪S\_MDF.MDF\_UID

**S\_USER (Responsible Person of the contract)**

X\_RESPONSIBLE\_PERSON🡪S\_USER.LOGIN

**S\_DOC\_AGREE (Original Contract ID)**

**Fk to S\_DOC\_AGREE on X\_ORIGINAL\_CONTRACT\_ID**

AGRXORIGINALAGREE0-🡪 S\_DOC\_AGREE .REV\_NUM

AGRXORIGINALAGREE2-🡪 S\_ORG\_EXT.NAME

AGR\_XORIGINALAGBI0🡪S\_BU.ROW\_ID

AGR\_XORIGINALAGREE 🡪 S\_DOC\_AGREE .NAME

AGR\_XORIGINALAG\_BI 🡪 S\_BU.ROW\_ID

**Fk to S\_PROD\_INT**

AGR\_PROD\_BI🡪S\_BU.ROW\_ID

AGR\_PROD\_NAME🡪 S\_PROD\_INT.NAME

AGR\_PROD\_VEN\_BI🡪 S\_BU.ROW\_ID

AGR\_PROD\_VEN\_LO🡪S\_ORG\_EXT.LOC

AGR\_PROD\_VEN\_NAME🡪S\_ORG\_EXT.NAME

**S\_DOC\_AGREE (Parent Agreement ID)**

**FK to S\_DOC\_AGREE on PAR\_AGREE\_ID**

PAR\_AGREE\_REVNUM 🡪 S\_DOC\_AGREE .REV\_NUM

PAR\_AGREE\_ACC\_NAME 🡪 S\_ORG\_EXT.NAME

PAR\_AGREE\_ACC\_BI 🡪S\_BU.ROW\_ID

PAR\_AGREE\_NAME 🡪 S\_DOC\_AGREE .NAME

PAR\_AGREE\_BI 🡪 S\_BU.ROW\_ID

**S\_DOC\_AGREE (Template Agreement ID)**

**FK to S\_DOC\_AGREE on X\_TEMPL\_ID**

AGR\_X\_TEMPL\_ID **🡪** S\_DOC\_AGREE.ROW\_ID

### *Agreement Foreign Key Check*

S\_DOC\_AGREE.NAME

In order to check if an agreement exists based on provided User keys, the following rules apply:

* Lookup the agreement in S\_DOC\_AGREE using only AGREE\_IDENTIFIER, AGR\_BU as conditions.
* If only one agreement is returned, continue the loading using the returned record *(no need to check if TARGET\_ACCNT\_NAME match to the account of the returned agreement).*
* If multiple agreements are returned, compare the returned records with the TARGET\_ACCNT\_NAME and AGR\_REV\_NUM provided.
  + If one agreement matches all provided user keys, continue the loading using the returned record and raise a warning.
  + If no agreement matches, reject the line and raise an FK error.

S\_DOC\_AGREE.X\_ORIGINAL\_CONTRACT\_ID

In order to check if an agreement exists based on provided User keys, the following rules apply:

* Lookup the agreement in S\_DOC\_AGREE using only ORIGINAL\_AGR\_IDENTIFIER and BU (from flat file) as conditions.
* If only one agreement is returned, continue the loading using the returned record *(no need to check if ORIGINAL\_TARGET\_ACCNT match to the account of the returned agreement).*
* If multiple agreements are returned, compare the returned records with the ORIGINAL\_TARGET\_ACCNT and BU (from flat file) provided.
  + If one agreement matches all provided user keys, continue the loading using the returned record and raise a warning.
  + If no agreement matches, reject the line and raise an FK error.

S\_DOC\_AGREE.PAR\_AGREE\_ID

In order to check if an agreement exists based on provided User keys, the following rules apply:

* Lookup the agreement in S\_DOC\_AGREE using only PAR\_AGREE\_IDENTIFIER and BU (from flat file) as conditions.
* If only one agreement is returned, continue the loading using the returned record*.*
* If multiple agreements are returned, compare the returned records with the TARGET\_ACCNT\_NAME and BU (from flat file) provided.
  + If one agreement matches all provided user keys, continue the loading using the returned record and raise a warning.
  + If no agreement matches, reject the line and raise an FK error.

***Note:*** *Parent Agreement should already exist in DB. If the parent agreement is in the same flat file as the child agreement, the child agreement will be rejected with errors.*

S\_DOC\_AGREE.X\_TEMPL\_ID

In order to check if an agreement exists based on provided User keys, the following rules apply:

* Lookup the agreement in S\_DOC\_AGREE using only TEMPLATE\_IDENTIFIER and BU (from flat file) as conditions.
* If only one agreement is returned, continue the loading using the returned record*.*
* If multiple agreements are returned, reject the line and raise an FK error.

## Other Validations

### Managing primary contract account:

* For each contract (user key AGREE\_IDENTIFIER & AGR\_BU) at least one account linked to it (TARGET\_ACCNT\_NAME) needs to be set as primary i.e. X\_PRIMARY\_ACCNT\_FLG set to “Y”.
* Use following logic to set primary account :
  + Check per AGREE\_IDENTIFIER & AGR\_BU combination that exactly one account has X\_PRIMARY\_ACCNT\_FLG set to “Y”.
    - If this is OK 🡪 Load flat file as is and set account with the flag set to Y as primary
    - If > 1 account has the flag set to Y, reject all records for that contract.
    - If no account has the flag set to Y, reject all records for that contract.

### Managing child contracts

* For each contract (user key AGREE\_IDENTIFIER & AGR\_BU) with an account linked to it (TARGET\_ACCNT\_NAME) and which is not set as primary (Child Contract) i.e. X\_PRIMARY\_ACCNT\_FLG set to “N”.

Use following logic to set primary account:

* + Check the AGREE\_IDENTIFIER & AGR\_BU which has X\_PRIMARY\_ACCNT\_FLG set to “Y” (Primary Contract).
    - In errors table (WC\_ERRORS\_G) if error has been logged for parent record.
    - And if the record is present in base table or not

IF Primary contract has errored out + Primary contract **is not** present in base table THEN **reject** record

IF Primary contract has errored out + Primary contract **is** present in base table THEN **do not** reject record

### ACTIVE\_FLG and Status

If the AGR\_ACTIVE\_FLG is set to Y when the AGR\_STAT\_CD is not Active then an error will be output so that the records with this discrepancy will be rejected accordingly.

### Cancellation Reason & Cancellation Detail

The following validation applies to the above field in the loader:

1. If Agreement status (AGR\_STAT\_CD) is not in (‘Pending,’Cancelled), Cancellation Reason is not Mandatory and should be Null.
2. If Agreement status (AGR\_STAT\_CD) = ‘Cancelled’, Cancellation Date is Mandatory should be populated.

### Validation on Contracted Start Date

1. Contracted Start Date (X\_CONTRACTED\_START\_DT) should be less than Contracted End Date (X\_CONTRACTED\_END\_DT)

# Additional Functionality

## Fixed template flag processing

* The AGR\_X\_FIXEDTMPLFLG (Fixed template flag) will not be set for the KA Agreements

## Additional Notes on field AGR\_X\_FIXEDTMPLFLG

If the X\_FIXED\_TMPL\_FLG on the contract is set to “Y”, CI records will be rejected for all markets except Russia whereby we have an additional check as shown below:

|  |  |  |  |
| --- | --- | --- | --- |
| CI Loader | | | |
| Condition 2 | | | |
| ***Parent Contract FIXED\_TMPL\_FLG*** | ***Parent Contract X\_TMPL\_ID*** | ***ITM\_AGREE\_BU*** | ***CI can be loaded?*** |
| NULL | <ANY VALUE> | All markets | Y |
| N | <ANY VALUE> | All markets | Y |
| Y | <ANY VALUE> | All markets but Russia | N |
| Y | <ANY VALUE> | Russia, RU | Y |

# IFB Files

The IFB Files for this entity are

**Insert/Updates**

IFA\_XXX\_CH\_Contract.ifb

Note: XXX stands for the country code i.e ESN, ITA

# Error Messages

Following error messages could be raised by the interface due to validations implemented:

|  |  |  |
| --- | --- | --- |
| **Error message** | **Root Cause** | **How to fix it** |
| The Foreign Key does not exist | No record exist in the DB to link the record to | Validate values provided on the corresponding FK columns and provide delta file with correct ones for reload. |
| The corresponding position does not exist | Position to link the account to doesn’t exist | Validate values provided on the corresponding FK columns and provide delta file with correct ones for reload. |
| Multiple records have the same Agreement Unique Identifier. ***(Warning)*** | S\_DOC\_AGREE.NAME is not unique | Record will continue to load. Market should be informed of the duplicate S\_DOC\_AGREE.NAME. |
| More than one record matches user keys. | All user keys provided for Agreement resolve to multiple records (Conflict ID differs) | Record will be rejected. |

# Example of flat file

# Interface execution

To execute this interface it would be needed to follow the steps below:

* Place the CHXX file on server path XXXX.
* Execute the worklet RUN\_EIM\_PROCESS\_GENERIC.wrk\_pre\_Etl from EIM\_INTERFACE folder
* Execute the worklet RUN\_EIM\_PROCESS\_GENERIC.wrk\_Contract\_CH\_Interface\_Generic from EIM\_INTERFACE folder
* Execute the worklet RUN\_EIM\_PROCESS\_GENERIC.wrk\_Export\_Errors from EIM\_INTERFACE folder
* Validate records loaded on EIM\_AGREEMENT table for IF\_ROW\_BATCH\_NUM = xxxxx
* Validate Error file generated on server path XXXX.
* Validate records updated/created from the OBIEE / TME Sales application.