

Data Flow between JDE and APS system

This Document focuses on the first steps of the process and involves the tables associated with JD Edwards.

Flow:

- Flow 1.** Data comes in flat files from the **SHOP Floors** through CO-ACT applications.
- Flow 2.** These inbound data is loaded into the **Base Tables** like BOM (Billing of Material), Routing, Sourcing, Item etc.
Here some tables are truncated and then the data is loaded and in case of other tables, the data is only appended to the existing one.
- Flow 3.** For each insertion into the Base tables, there are predefined triggers which load the same data into the **K tables**.
- Flow 4.** Through some integration jobs, e.g. ETHJP306, data is also loaded into the **C tables** and at the same time K tables are truncated.

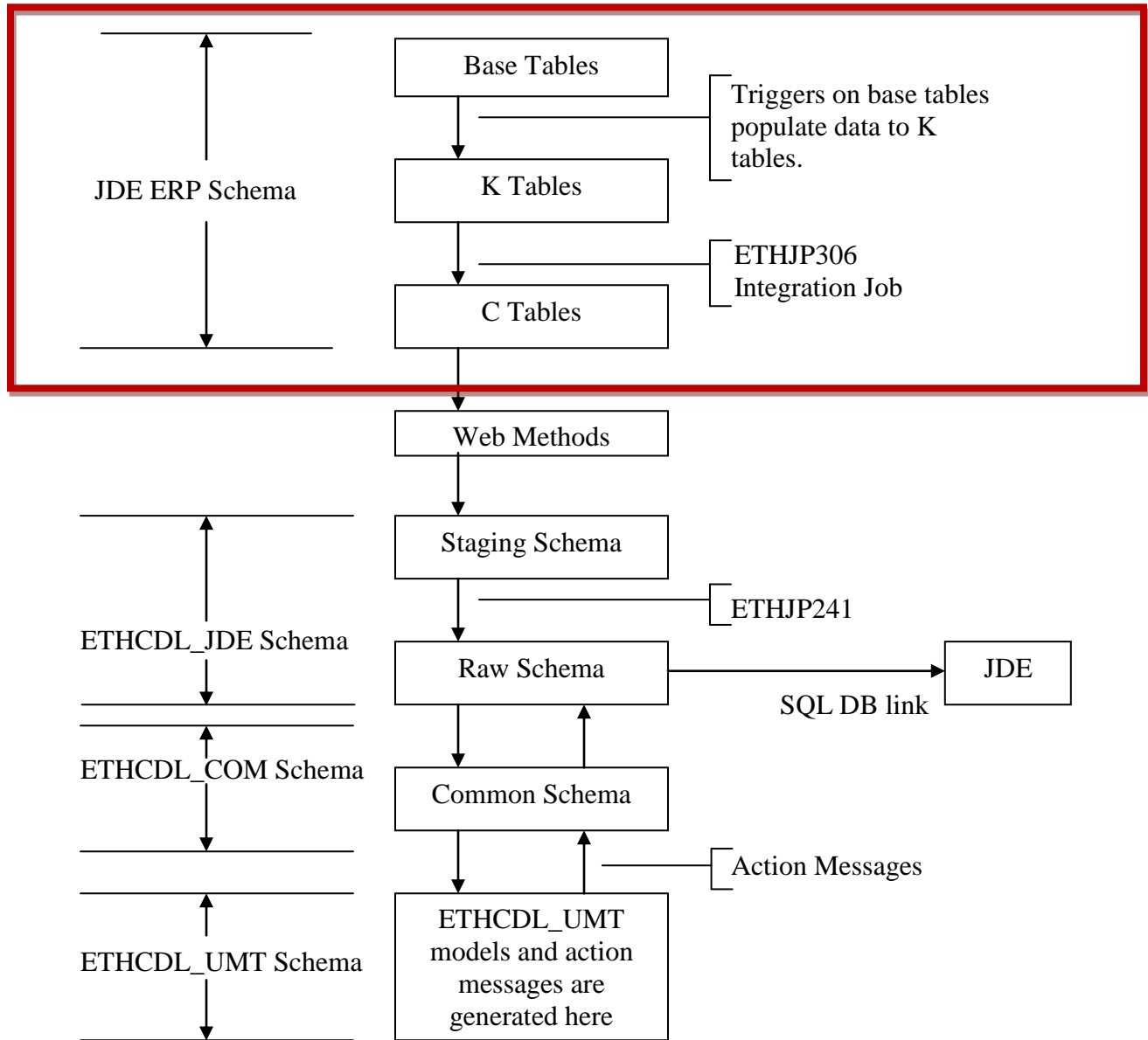
The standard naming convention for the tables:

If the Base table name is F3002, then the corresponding K table name is F593002K and the C table name is F593002C.

- Flow 5.** From the C tables, data comes into the **Staging Schema** through **WEB Methods**, which is out of scope of APS. From here the job **JP241** loads data into the **RAW schema (ETHCDL JDE)** maintaining a 1:1 mapping between tables of Staging and Raw schema.
- Flow 6.** From Raw schema (ETHCDL_JDE), data goes to **Common schema (CDL COM)**. A full validation is done by PL/SQL processes before the data is loaded into the COMMON schema (ETHCDL_COM).
- Flow 7.** There are jobs in **User schema (ETHCDL UMT)**, which generates models and action messages with the data present in COMMON schema (ETHCDL_COM).
- Flow 8.** These actions messages, output of the modeling runs, travel the same path in the reverse order, in which the data has originally entered the system. From the COMMON schema (ETHCDL_COM) tables, data is transferred

back to RAW schema (ETHCDL_JDE) tables, through a PL/SQL process and from there into the JDE via SQL db link.

DATA FLOW DIAGRAM:



Base Tables

Base tables are standard JDE Tables. A **trigger** exists for these tables that copies each record into its correspondent K table with adequate format –fields are truncated, some are converted, etc.- . Triggers also act when a record is updated or deleted, impacting in the K table as well. Each action has a code that is inserted in the TNAC field of the K table.

K tables

K Tables are also called **Key Tables**. Records can be inserted, updated or deleted from the Base Table. The trigger reflects this in the TNAC field with codes for each action:

- INSERT (“Add”) = 'A'
- UPDATE (“Change”) = 'C'
- DELETE = 'D'

*For example, a record is inserted in F4101. The **trigger** inserts a version of it in its K table, F594101K. The TNAC field will have 'A' for Add. Then, record is updated in the Base Table. The TNAC of the K table will be 'C' now.*

From K tables to C tables

Records from K tables are transferred to its corresponding C (“**Buffer Tables**”) table in two ways: automatic –by the same UBE, etc. that inserted them- and scheduled by **ETHJP306 Integration Job**. This Job executes the `cai_e1Extract.extract_from_key_tables` package in the Database. A stored procedure then processes each record from K to C; *this is reflected in the PID field being something like 'SP01/17/12.'* Records prompted by the Database during this process are retrieved and inserted in the IMERN field. Although only a number followed by “-” is stored, the original error message contains “ORA-XXXX”. For example, for an IMERN = ‘-6502’, the error given was “ORA-6502”. Finally, the IMEDSP field shows how the process left the record, the codes are:

- Error = 'E'
- Processed = 'P'
- Ignore = 'I'

*Each K table has a different stored procedure for processing each row into the C tables; some log each action into the **F590001C** table. This is a **log** table and contains error status, error codes and description for each processed row.*