EWPS Load CE Data Module

**Revision History**

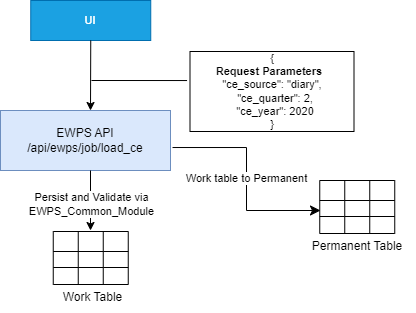
|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Latest Version Number** | **Brief Description of the Changes**  (Ex. Updates from Reqt. Review) | **Person Who Last Made Changes to the Document** |
| 11/10/2023 | 1.0 | Initial draft for EWPS Load CE Data Module | Kristopher Nguyen |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**DOCUMENT SCOPE**

This document describes the design and architecture of the Load CE Data Module. Anything other than Loading CE data is outside the scope of this document.

**Introduction**

Product owners want to design a module where they can load CE data in the form of SAS dataset into Oracle database. The initiation of the load job is via an API call from a UI client. After loading data successfully, all CE dataset must be deleted from the server where the parsing process took place. All data from SAS dataset must be met all database constraint and exist in the final permanent table. Below is the general flow of how Load CE module works:



**Prerequisites**

The following requirements must be met before initiate Load CE Data job:

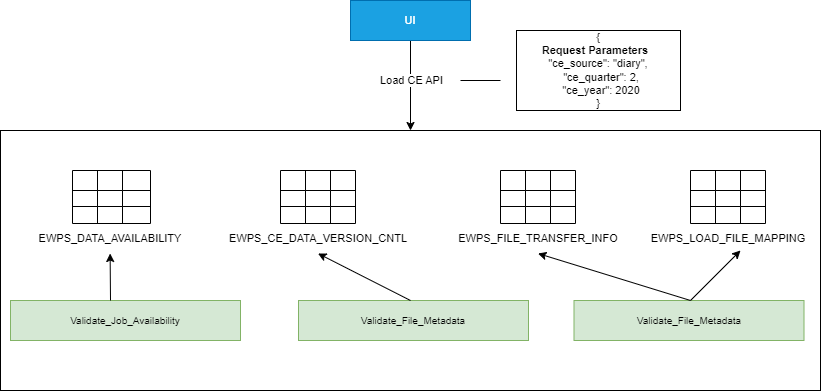
1. Transfer CE Data job must happen before initiate loading job. After transfer job complete, a set of artifacts must be existed on a designated CE storage server. This server location is configured via the application settings.
2. A set of CE data column, EWPS column, and EWPS table mapping must be existed in EWPS\_LOAD\_FILE\_MAPPING table.
3. Availability flag must be set to ‘Y’ for the transfer job record.

**Architecture Overview**

Load CE Module is a small part to the whole EWPS API eco system. In order to process a load job, a call is being made via a Client (EWPS Blazor UI). The request parameter is in the form of JSON object. (See Data Model for details). In order to be more UI responsive, a set of validations are performed and return the result quickly to the client.

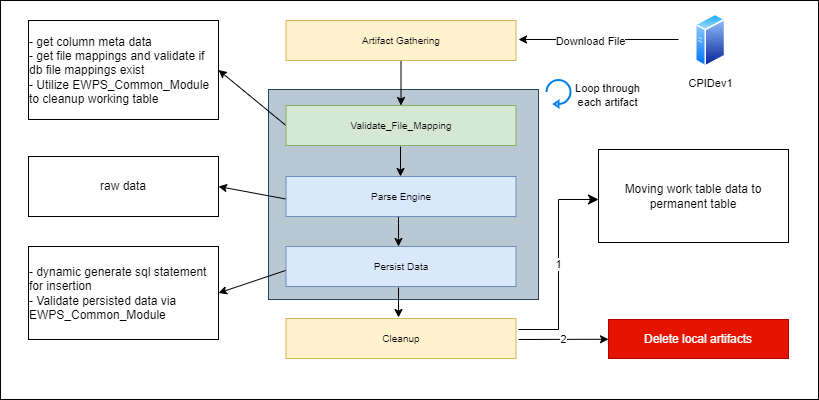
Up-front Validations:

1. **validate\_job\_availability**: there should be one transfer job made available to load. If “data\_availability\_fl” is set to “N”, return an error message to the client immediately.
2. **validate\_file\_metadata**: this validation is based on records stored in the database. The transfer job uses EWPS\_FILE\_TRANSFER\_INFO table for download information. After each download, the process record file meta information to the EWPS\_CE\_DATA\_VERSION\_CNTL table. The information in these two tables are being used to validate if transfer info matches with the version control table. If this information is mismatched, the load process will return an error immediate to the client.
3. **validate\_file\_mapping**: return validation error if the number of file does not matched to the number of table stored in the EWPS\_LOAD\_FILE\_MAPPING.



If all validations are successfully, the system will create a background job to run this long process job. Each step of the process will be log to the process log file. Below is the general flow of the background job.

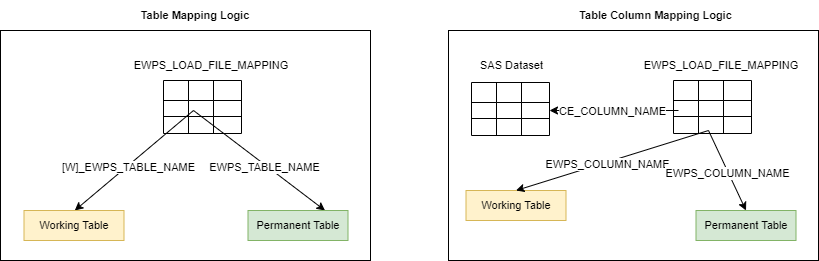
1. **Artifact gathering**: getting all files from the transfer jobs. These files are being stored in a specific CE Data storage compliance servers. The location of each … see EWPS\_FILE\_TRANSFER/Write\_File\_Path for the location of each file
2. **Validate\_File\_Mapping**: this step is crutial before parsing the SAS file. Utilize EWPS\_Common\_Module database package to perform database truncate and disable table constraints.
3. **Parse & Persist data**: parse each SAS and persist only mapped columns. After data is persisted in the work table, utilize EWPS\_Common\_Module to re-enable constraints. All errors will be saved in the EST\_Exception table.
4. **Cleanup**: Current process will be terminated if there are any errors reported inside EST\_Exception table. Otherwise, proceed to move data from work table to permanent table. If this process succeeds, the load process is completed.



**Data Model**

Each SAS dataset file is associated with a work and permanent oracle table. The mapping for this is stored in the EWPS\_LOAD\_FILE\_MAPPING under EWPS\_TABLE\_NAME column.

Each column in the SAS dataset is also mapped to particular column in the mapped work/permanent table. SAS Column is associated with CE\_COLUMN\_NAME column, EWPS\_COLUMN\_NAME is where the value will be stored in the database table.



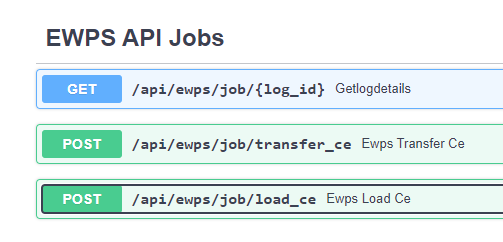
**Error Handling**

While load CE module is running, each primary step of the load processing will be stored in the process log file. The location of this file is configured via environment setting file – logging\_location. Any exception encountered will also be saved in this file along with the exception stack trace.

**Documentation for API**

Load Module API documentation is via Swagger. This tool will be automatically enabled in devl/syst/acct environments.

**Swagger URL**: <https://ewpsapidev.psb.bls.gov/>



**Testing Strategy**

The following test cases are used to test the Load module implementation. There are two set of test suites that can perform either manually or automatically using unit tests. This section will be on-going updated as we find out more about the nature of the requirements.

Test via Swagger tool or Unit Test:

**Up-front Validation**:

* Set Availability Flag to ‘N’, invalid ce\_source, or quarter will trigger the following error when call API via Swagger or Unit Test
* **{**
* **"has\_error": true,**
* **"data": null,**
* **"message": "Failed to validate current job.",**
* **"message\_details": [**
* **{**
* **"has\_error": true,**
* **"message": "No job found for the given quarter start and ce source."**
* **}**
* **]**
* **}**
* Calling API with invalid parameters:

Error: Unprocessable Entity

* **{**
* **"detail": [**
* **{**
* **"loc": [**
* **"body",**
* **"ce\_year"**
* **],**
* **"msg": "field required",**
* **"type": "value\_error.missing"**
* **}**
* **]**
* All SAS files are correctly mapped to work/permanent tables.
* All columns are correctly mapped to work/permanent column table.
* EWPS\_CE\_DATA\_VERSION\_CNTL contains number of files less than in the EWPS\_TRANSFER\_INFO table.

**Authentication**:

* Call API with a valid JWT Token. The result will be:
* **{**
* **"detail": "Invalid access token!"**
* **}**

**Background job**:

* If all up-front validations are successfully, a background job will be created with specific status. See table below for details.

|  |  |  |
| --- | --- | --- |
| JOB\_STATUS | JOB\_STATUS\_DESC | Note |
| 0 | PENDING | Beginning of background job |
| 1 | IN\_PROGRESS | This status remains throughout the life cycle of the load process |
| 3 | COMPLETE | Successfully load all files |
| 10 | ERROR | Anytime an error occurs |

* Remove at least one file from source server. Exception will be caught and log in process log file and job status should switch to error.

**EWPS\_Common\_Module**:

* Add duplicate rows to at least one SAS file should be recorded in Process Log and also in EST\_Exception table.
* Remove at least one mapped column in the EWPS\_LOAD\_FILE\_MAPPING should produce error and record in Process Log
* After load process complete, all work tables need to be empty, and constraint need to be enabled.