**Load Control Documentation**

**Process**

**Data Model**

**Rules**

# **Index**

[**Solution Overview 4**](#_heading=h.30j0zll)

[Requirements 4](#_heading=h.3znysh7)

[**The Load Control Process 5**](#_heading=h.2et92p0)

[Load Control Data Model 7](#_heading=h.tyjcwt)

[The Data Model Diagram 8](#_heading=h.3dy6vkm)

[**Load Control IICS Objects 9**](#_heading=h.1t3h5sf)

[Taskflow TSKF\_EDW\_INITIAL\_SETUP\_LOAD\_CTL\_2 10](#_heading=h.2s8eyo1)

[∙](#_heading=h.17dp8vu) Restart Process 10

[Mapping Task MTT\_INITIAL\_VALIDATION\_LOAD\_CTL\_2\_1 10](#_heading=h.3rdcrjn)

[Mapping Task MTT\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2 11](#_heading=h.26in1rg)

[Any EDW Taskflow Setup 12](#_heading=h.lnxbz9)

[∙](#_heading=h.35nkun2) Restart Process 12

[Mapping Task MTT\_INITIAL\_TASKFLOW\_LOAD\_CTL\_4\_1 12](#_heading=h.1ksv4uv)

[Mapping Task MTT\_END\_TASKFLOW\_LOAD\_CTL\_4\_2\_2 14](#_heading=h.44sinio)

[Taskflow TSKF\_EDW\_END\_LOAD\_CTL\_5 14](#_heading=h.2jxsxqh)

[∙](#_heading=h.z337ya) Restart Process 15

[**The Global Parameter 16**](#_heading=h.3j2qqm3)

[**Including Load Control in EDW Taskflows 17**](#_heading=h.1y810tw)

[Configuration at Taskflow Level 17](#_heading=h.4i7ojhp)

[Load Control Mapping Task Configuration in EDW Taskflows 17](#_heading=h.2xcytpi)

[∙](#_heading=h.1ci93xb) MTT\_INITIAL\_TASKFLOW\_LOAD\_CTL\_4\_1 17

Version Control

| **Version** | **Date** | **Change** | **Author** | **Review Date** |
| --- | --- | --- | --- | --- |
| 1.0 | 2/14/2022 | Document Creation | Rafael Fonseca | 2/25/2022 |
|  |  |  |  |  |

**Solution Overview**

The Load Control solution will allow to the customer handle the load control for its Data Integration processes based on Informatica Intelligent Cloud Services CDI. Also, will provide visibility and capabilities to scale and add new parameters. This document will provide information and details about Load Control Process, Data Model, IICS Objects and recommendation for Taskflow Design for EDW and how to integrate the Load Control components and Restart process for Jobs.

**Requirements**

Provide capability to configure and parametrize the Enterprise Data Warehouse load. All the EDW Data Integration is set to operate using parameters. Some parameters are fixed value and others are logic-based populated. Also, there is a load control table that will allow to track all executions on Data Warehouse. Customer will be able to define new parameters and use them in the existing and new tasks.

**The Load Control Process**

The process is divided in 5 major steps considering the EDW load will be started from Control-M.

The first step is Control-M starting the Load Control Setup Taskflow (LCST). Then LCST (second step in the process) starts and performs validations against the Load Control Table to make sure there is no execution plan still running before generating a new Execution Plan. If the validation fails, the LCST will be suspend. Otherwise, the LCST will backup the last parameter file and generate a new Execution Plan (Load\_Status = ‘R’) and parameter file.

Diagram

Description automatically generated

If any failure occurs during the parameter file generation or the generation of the new Execution Plan, the LCST will be suspended. Otherwise, the Control-M will start the EDW Taskflows according to the defined Sequence initiating a loop from Step 3 to Step 4.3.2 until it processes the last EDW Taskflow so the Load Control can update the Execution Plan (Load\_Status = ‘C’) in the Step 5.

Diagram

Description automatically generated

For more details about the Load Control Process, please refer to the *EDW Load Control Process.vsdx* document and its PDF version.

**Load Control Data Model**

The Data Model contains 3 tables in the following. For the Data Model details and modification, please refer to the *EDW Load Control.pdm* PowerDesign Physical Data Model file.

All the load control tables reside in Snowflake developed under the EDW\_DEV database and EDW\_LOAD\_CT schema.

The following is the description of each Load Control table:

**IICS\_EDW\_LOAD\_SETUP\_CTL:** Stores the Execution Plan for every load and holds the status in which can be Running or Completed.

| **Table Dictionary** | | | | |
| --- | --- | --- | --- | --- |
| # | **Field Name** | **PK/FK** | **Data Type** | **Description** |
| 1 | CTL\_LOAD\_ID | PK | BIGINT | Stores the Unique information about the Load. Each load has a unique CTL\_LOAD\_ID. It is a sequence. |
| 2 | ETL\_PROC\_WID | N | VARCHAR(256) | It is a continuation sequence from the last value of $$ETL\_PROC\_WID parameter. |
| 3 | INTEGRATION\_ID | N | VARCHAR(256) | It is a continuation sequence from the last value of $$INTEGRATION\_ID parameter. |
| 4 | LOAD\_DT | N | TIMESTAMP | Information of the Extraction Date that the Parameter File will contain. This is a calculated field, usually the continuation of the previous date. |
| 5 | START\_TS | N | TIMESTAMP | The Date/time when the Execution Plan has been Created. |
| 6 | END\_TS | N | TIMESTAMP | The Date/time when the Execution Plan has been Completed. |
| 7 | LOAD\_STATUS | N | CHAR(1 BYTE) | Status of the execution ('R' for Running and 'C' for Completed). |
| 8 | LAST\_UPDATE\_DATE | N | TIMESTAMP | Stores the date where the record has been modified in the table. |

**IICS\_EDW\_LOAD\_TASKFLOW\_CTL:** Stores the Execution Plan information plus the EDW Taskflow Name in wich is running. Each EDW Taskflow will have a Load Control Mapping task in its beginning (to generate the row before running any EDW Mapping Task) and in the end that will be responsible to update this table to indicate that a given EDW Taskflow has completed.

| **Table Dictionary** | | | | |
| --- | --- | --- | --- | --- |
| # | **Field Name** | **PK/FK** | **Data Type** | **Description** |
| 1 | TASKFLOW\_LOAD\_ID | PK | BIGINT | Stores the Unique information about the EDW Taskflow Load. Each load has a unique TASKFLOW\_LOAD\_ID. It is a sequence. |
| 2 | CTL\_LOAD\_ID | PK,FK | BIGINT | Foreign Key that references to the IICS\_EDW\_LOAD\_SETUP\_CTL table. |
| 3 | ETL\_PROC\_WID | N | VARCHAR(256) | Optional Field. Redundant information from the Setup Load Control Table. |
| 4 | INTEGRATION\_ID | N | VARCHAR(256) | Optional Field. Redundant information from the Setup Load Control Table. |
| 5 | PROJECT\_NAME | N | VARCHAR(256) | Optional and still not supported as Variable so it cannot be fetched at mapping level. |
| 6 | FOLDER\_NAME | N | VARCHAR(256) | Optional and still not supported as Variable so it cannot be fetched at mapping level. |
| 7 | TASKFLOW\_NAME | N | VARCHAR(256) | Name of the EDW Taskflow. |
| 8 | LOAD\_DT | N | TIMESTAMP | Optional. Redundant information that already exists in the IICS\_EDW\_LOAD\_SETUP table. |
| 9 | START\_TS | N | TIMESTAMP | The Date/time when the EDW Taskflow has started. |
| 10 | END\_TS | N | TIMESTAMP | The Date/time when the EDW Taskflow has been Completed. |
| 11 | LOAD\_STATUS | N | CHAR(1 BYTE) | Status of the execution ('R' for Running and 'C' for Completed). |
| 12 | LAST\_UPDATE\_DATE | N | TIMESTAMP | Stores the date where the record has been modified in the table |

**IICS\_EDW\_LOAD\_DETAILS\_CTL: Not supported for the current phase**. This table would store the Mapping Task Details for each EDW Taskflow. This is not implemented as the Load Control Granularity works at Taskflow level, so it is not supported capturing information from each Mapping Task and report it in this table. To be implemented in the future, possibly using Rest/APIs.

| **Table Dictionary** | | | | |
| --- | --- | --- | --- | --- |
| # | **Field Name** | **PK/FK** | **Data Type** | **Description** |
| 1 | MTT\_LOAD\_ID | PK | BIGINT | Stores the Unique information about the EDW MTT Load. Each load has a unique MTT\_LOAD\_ID. It is a sequence. |
| 2 | TASKFLOW\_LOAD\_ID | PK | BIGINT | Foreign Key that references to the IICS\_EDW\_LOAD\_TASKFLOW\_CTL table. |
| 3 | CTL\_LOAD\_ID | PK,FK | BIGINT | Foreign Key that references to the IICS\_EDW\_LOAD\_TASKFLOW\_CTL table. |
| 4 | ETL\_PROC\_WID | N | VARCHAR(256) | Optional Field. Redundant information from the Setup Load Control Table. |
| 5 | INTEGRATION\_ID | N | VARCHAR(256) | Optional Field. Redundant information from the Setup Load Control Table. |
| 6 | PROJECT\_NAME | N | VARCHAR(256) | The Project where the MTT resides. |
| 7 | FOLDER\_NAME | N | VARCHAR(256) | The Folder where the MTT resides. |
| 8 | TASKFLOW\_NAME | N | VARCHAR(256) | Name of the EDW Taskflow. |
| 9 | MTT\_NAME | N |  | Name of the EDW Mapping Task. |
| 10 | LOAD\_DT | N | TIMESTAMP | Optional. Redundant information that already exists in the IICS\_EDW\_LOAD\_SETUP table. |
| 11 | START\_TS | N | TIMESTAMP | The Date/time when the EDW Taskflow has started. |
| 12 | END\_TS | N | TIMESTAMP | The Date/time when the EDW Taskflow has been Completed. |
| 13 | TOTAL\_SUCCESS\_READ\_ROWS | N | BIGINT | The total amount of success read rows from all sources. |
| 14 | TOTAL\_FAILED\_READ\_ROWS | N | BIGINT | The total amount of failed read rows from all sources. |
| 15 | TOTAL\_SUCCESS\_TARGET\_ROWS | N | BIGINT | The total amount of success written rows into all targets. |
| 16 | TOTAL\_FAILED\_TARGET\_ROWS | N | BIGINT | The total amount of failed rows into all targets. |
| 17 | LOAD\_STATUS | N | CHAR(1 BYTE) | Status of the execution ('R' for Running and 'C' for Completed). |
| 18 | LAST\_UPDATE\_DATE | N | TIMESTAMP | Stores the date where the record has been modified in the table. |

**The Data Model Diagram**

A picture containing graphical user interface

Description automatically generated

**Load Control IICS Objects**

The Load Control relies in the IICS technology therefore it is made by Mappings, Mapping Tasks, Taskflows and Connection. All the Cloud Data Integration objects reside in the EDW\_LOAD\_CONTROL project.

**Graphical user interface, text, email

Description automatically generated**

Graphical user interface, text, application

Description automatically generated

Every IICS ETL Object like Mapping/Mapping Task and Taskflow have a name convention where it will identify the Step correspondent to the Load Control Process.

The following is the description of each Load Control Object by the Process Step Order:

**Taskflow TSKF\_EDW\_INITIAL\_SETUP\_LOAD\_CTL\_2**

It is the First Load Control Taskflow that is invoked by Control-M before any EDW Load starts. It represent the whole Step 2 in the Load Control Process.

**Type:** Taskflow

* **Recovery Process**

1. When any fail occurs this Taskflow is going to get Suspended Status.
2. Control-M Job will get a failed status and an email notification will be sent to the BI Team.
3. The Failed Taskflow will be Resumed by the Operator/BI Team. (to confirm who).
   1. The Operator/BI Team will Request Force-OK for the Control-M Job after the Taskflow Successfully completed.

A picture containing diagram

Description automatically generated

This taskflow is responsible for the validations against the Load Control Tables, Parameter File Backup and for generating the Execution Plan and the new Parameter file for the current load.

**Mapping Task MTT\_INITIAL\_VALIDATION\_LOAD\_CTL\_2\_1**

It is the First Load Control Mapping Task that runs in the taskflow TSKF\_EDW\_INITIAL\_SETUP\_LOAD\_CTL\_2. It is responsible for validating that there is no Execution Plan already running. If there is any Execution Plan running this Mapping Task will fail and the Taskflow will get Suspended.

**Associated Mapping:** m\_INITIAL\_VALIDATION\_LOAD\_CTL\_2\_1

Diagram

Description automatically generated

**Mapping Task MTT\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2**

This Mapping Task is responsible for backuping the last parameter file (in the Pre-Command), calculating parameters, generating the execution plan and new parameter file.

**Associated Mapping:** m\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2

Diagram

Description automatically generated

This mapping calculates the dynamic parameters and generates a new global parameter file and inserts a record in the Load Control Setup table in order to generate an Execution Plan.

In order to calculate dynamic parameters **the EXP\_CALCULATE\_PARAMETERS** is used. It uses variable as base for calculating dynamic parameters. Also there are some additional validation rules in which can make this mapping failing in case it finds any discrepancy in the Load Control Table. For all new parameter or even maintenance of existing dynamic parameters this expression transformation is the place where parameters are set. Graphical user interface, text

Description automatically generated with medium confidence

**Any EDW Taskflow Setup**

Each EDW Taskflow is the group of EDW Mapping tasks sequenced on a way it will respect the proper order and data integrity. For every EDW Taskflow, it is necessary to include the following Load Control Mapping tasks:

1. MTT\_INITIAL\_TASKFLOW\_LOAD\_CTL\_4\_1
2. MTT\_END\_TASKFLOW\_LOAD\_CTL\_4\_2\_2

Every EDW Taskflow structure will be compose by a Load Control Mapping task in the beginning, EDW Mapping tasks (according their sequence) and a Load Control Mapping task in the end. On the start a given EDW Taskflow will run first a Load Control Mapping task to generate a proper record in the Taskflow Load Control table to indicate that specific EDW Taskflow started running (LOAD\_STATUS = R), then all EDW Mapping tasks will run. After all EDW Load Control Mapping task run successfully, the last Mapping task (CTL\_4\_2\_2) will run to update the generated row by CTL\_4\_1 Load Control Mapping Task to have LOAD\_STATUS=C to indicate the EDW Taskflow has ran successfully.

The following is a EDW Taskflow Example (not real because in the present time the documentation is edited there was no EDW Taskflow built):

Chart, waterfall chart

Description automatically generated

More details about how to include the Load Control Mapping Tasks in the EDW Taskflow will be provided later in this document.

* **Restart Process**

1. When any fail occurs this Taskflow is going to get Suspended Status.
2. Control-M Job will get a failed status.
3. The Failed Taskflow will be Resumed by the Operator.
4. The Operator will Request Force-OK for the Control-M Job after the Taskflow Successfully completed.

**Mapping Task MTT\_INITIAL\_TASKFLOW\_LOAD\_CTL\_4\_1**

This Mapping Task needs to be included in the beginning of every EDW Taskflow, because it is responsible to validate the Load Control Information and generate a record in the Taskflow Load Control Table, indicating that the EDW Taskflow is running (LOAD\_STATUS=R), if the Load Control Information is valid. In case of the Load Control Information is not valid it will fail or suspend the EDW Taskflow. This Mapping Task is configured to run in parallel as some EDW Taskflows could run in parallel so the Mapping Task will run in different Taskflows.

Reasons to the Load Control Information be invalid:

1. There is no Execution Plan Created in the **IICS\_EDW\_LOAD\_SETUP\_CTL** table.
2. The Execution Plan in the **IICS\_EDW\_LOAD\_SETUP\_CTL** table does not match to the generated parameter information (CTL\_LOAD\_ID does not match).
3. There is already a non completed record (LOAD\_STATUS = R) in the **IICS\_EDW\_LOAD\_TASKFLOW\_CTL** table for that Workflow.

**Associated Mapping:** m\_INITIAL\_TASKFLOW\_LOAD\_CTL\_4\_1

Diagram

Description automatically generated

The Mapping uses In-Out Parameters in for the following purposes:

1. Receive the Name of the EDW Taskflow (because there is no buit-in system variable to be used at Mapping level).
2. Read the Parameter File and compare the CTL\_LOAD\_ID information to validate the Load Control Information.

Graphical user interface, text

Description automatically generated

**Mapping Task MTT\_END\_TASKFLOW\_LOAD\_CTL\_4\_2\_2**

This Mapping Task needs to be included in the end of every EDW Taskflow, because it is responsible update the Taskflow Load Control Table, indicating that the EDW Taskflow has completed (LOAD\_STATUS=C). This Mapping Task is configured to run in parallel as some EDW Taskflows could run in parallel so the Mapping Task will run in different Taskflows.

**Associated Mapping:** m\_ END\_TASKFLOW\_LOAD\_CTL\_4\_2\_2

Text

Description automatically generated

The Mapping uses In-Out Parameters in for the following purposes:

1. Receive the Name of the EDW Taskflow (because there is no buit-in system variable to be used at Mapping level).
2. Receive the Decision Status in from the Taskflow Level to update LOAD\_STATUS=F if needed. But this functionality is not in use because it does not fit for Taskflows having more than one EDW Mapping task. The idea was use just one generic mapping to update the Taskflow Load Control Table to report Completion o Failure, but it is not in use.

Graphical user interface, text, application

Description automatically generated

**Taskflow TSKF\_EDW\_END\_LOAD\_CTL\_5**

It is the Last Load Control Taskflow that is invoked by Control-M after the last EDW Taskflow Load successful completes. It represent the whole Step 5 in the Load Control Process.

**Type:** Taskflow

This Taskflow is responsible to update the Execution Plan (generated in the Step 2) indicating that its execution has successfully completed and save the Calculated parameters in the Parameter Table. It is a Sequencial Taskflow having two Mapping Tasks:

1. MTT\_END\_SETUP\_LOAD\_CTL\_5\_0
2. MTT\_END\_SETUP\_LOAD\_CTL\_UPDATE\_PARAM\_5\_0

* **Restart Process**

1. When any fail occurs this Taskflow is going to get Failed Status.
2. Control-M Job will get a failed status.
3. The Failed Mapping Task(s) will be Restarted by the Operator.
4. The Operator will Request Force-OK for the Control-M Job after the Taskflow Successfully completed.

**The Global Parameter**

The Load Control process is responsible for generating the Global Parameter File in which will feed the EDW Taskflows and subsequent Load Control steps.

The Taskflow TSKF\_EDW\_INITIAL\_SETUP\_LOAD\_CTL\_2 has a mapping task (m\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2) that is responsible for generating the global parameter file. The mapping basically reads the parameter table in Snowflake database and calculates dynamic parameters and then generates a new parameter file in which will be the parameter file that will need to be used on each Load Control Mapping Task and EDW Mapping Task.

The following is the location in Snowflake where the parameter table resides:

**Database:** EDW\_DEV

**Schema:** EDW\_LOAD\_CT

**Table Name:** IICS\_EDW\_PARAM\_CTL

The first time when all the EDW Loads will start from IICS (no from PowerCenter anymore), it is important to load the last parameter file generated in PowerCenter so the parameter table can have all the parameters.

For that it is necessary to run the following Mapping to load the PowerCenter parameter into the Snowflake parameter table:

**Project:** EDW\_LOAD\_CONTROL

**Mapping Name:** m\_LOAD\_PC\_DAC\_PARAMETERS\_TO\_IICS\_SNOWFLAKE\_PARAMETERS

Graphical user interface, diagram, application

Description automatically generated

This mapping is one-time load mapping therefore there is no Mapping Task/Taskflow Associated with it. This mapping truncates the target (parameter table in Snowflake), so when running this mapping it is recommended to be careful.

The following steps are used to materialize parameters in snowflake parameter table, when the EDW Load will start using IICS, and not run from PC anymore.

1- Get the pc parameter file and curate it (eliminating duplicates from local/global)

2- Rename the curated parameter file to MASTER\_SOURCE\_OF\_PARAMETERS\_CTL.csv and include $$PRUNE\_DAYS=6 on the first row in the curated file

3 - Include the curated file MASTER\_SOURCE\_OF\_PARAMETERS\_CTL.csv in /INFA\_SHARED/SrcFiles

4 - Run the Mapping

Notice: The step 2 ($$PRUNE\_DAYS=6) is very important because it is used to calculate the $$LAST\_EXTRACT\_DATE parameter.

**Including Load Control in EDW Taskflows**

As explained previously in this documentation, every EDW Taskflow needs two Load Control Mapping Tasks:

1. MTT\_INITIAL\_TASKFLOW\_LOAD\_CTL\_4\_1 (in the beginning)
2. MTT\_END\_TASKFLOW\_LOAD\_CTL\_4\_2\_2 (in the end)

Chart, waterfall chart

Description automatically generated

**Configuration at Taskflow Level**

Every EDW Taskflow must have its Advanced configuration to Suspend on fault as screenshot below:

Graphical user interface, text, application

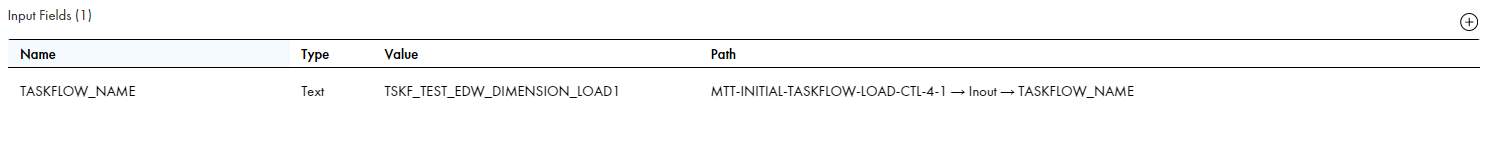
Description automatically generated

**Load Control Mapping Task Configuration in EDW Taskflows**

* **MTT\_INITIAL\_TASKFLOW\_LOAD\_CTL\_4\_1**

**Nomenclature:** This mapping task must be the first Mapping Task to run on every EDW Taskflow. It is recommended to give the same Mapping Task Name for the Data Task Name in the Taskflow.

**Input Field:** Due the IICS Limitation in the Cloud Mapping Designer by not having a system variable $PMTaskflowName (as PowerCenter has $PMWorkflowName), the Mapping Task needs to feed its In-Out Parameter ($$TASKFLOW\_NAME) in order to generate the row in the Taskflow Load Control Table with the proper Taskflow Name.

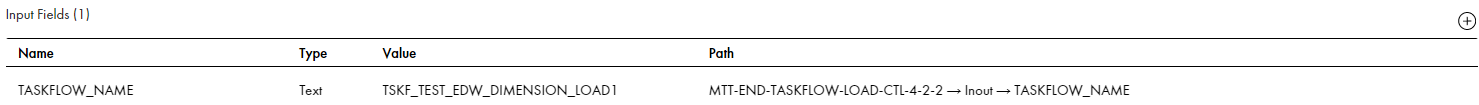


After this Load Control Mapping Task, all EDW Mapping Task must be configured in the EDW Taskflow by following the Sequence. The EDW Mapping tasks must be configured to run only if the first Load Control Mapping Task Successfully ran.

* **MTT\_END\_TASKFLOW\_LOAD\_CTL\_4\_2\_2**

**Nomenclature:** This mapping task must be the last Mapping Task to run on every EDW Taskflow. It is recommended to give the same Mapping Task Name for the Data Task Name in the Taskflow.

**Input Field:** Due the IICS Limitation in the Cloud Mapping Designer by not having a system variable $PMTaskflowName (as PowerCenter has $PMWorkflowName), the Mapping Task needs to feed its In-Out Parameter ($$TASKFLOW\_NAME) in order to update the row in the Taskflow Load Control Table with the proper Taskflow Name.



After the all EDW Mapping tasks successfully ran this Mapping Task must be positioned after the last EDW Taskflow.