IICS Parameters and Taskflow Orchestration

Summary

[Parameter File Generation 2](#_Toc107873362)

[Materialization of Parameters In Snowflake Parameter Table 2](#_Toc107873363)

[The generation of the global parameter file for the EDW Load 2](#_Toc107873364)

[Updating snowflake param table with latest parameter values 3](#_Toc107873365)

[Validation of Dynamic Parameter Values 4](#_Toc107873366)

[The transition from task flow to task-flow (Control-M related) 6](#_Toc107873367)

[Load Control Taskflow Table on Snowflake 8](#_Toc107873368)

[Failure and Recovery Process 9](#_Toc107873369)

[Fail to Recovery 9](#_Toc107873370)

[ETL process setup (Load control + EDW Taskflows) 11](#_Toc107873371)

[Starting off: 11](#_Toc107873372)

[First mapping of the taskflow01 (INITIAL MAPPING TASK): 12](#_Toc107873373)

# Parameter File Generation

## Materialization of Parameters In Snowflake Parameter Table

**Master Source Parameter -> param table at snowflake**

**Database:** EDW\_DEV

**Schema:** EDW\_LOAD\_CT

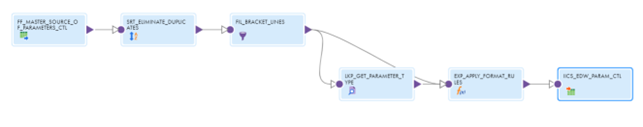
**Table Name:** IICS\_EDW\_PARAM\_CTL

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



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.

## The generation of the global parameter file for the EDW Load

The task flow TSKF\_EDW\_INITIAL\_SETUP\_LOAD\_CTL\_2 will load parameters from the snowflake param table to the global parameter file and create an execution plan

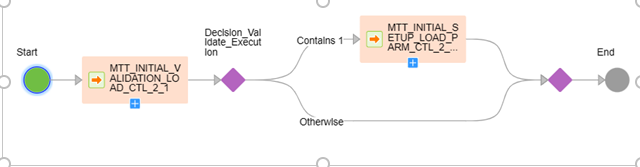
at snowflake setup table (CTL\_LOAD\_ID).

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

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 with all calculated parameters being calculated and loaded into the global parameter file along with static parameters.

param table at snowflake -> global parameter file at agent directory

TSKF\_EDW\_INITIAL\_SETUP\_LOAD\_CTL\_2:



## Updating snowflake param table with latest parameter values

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

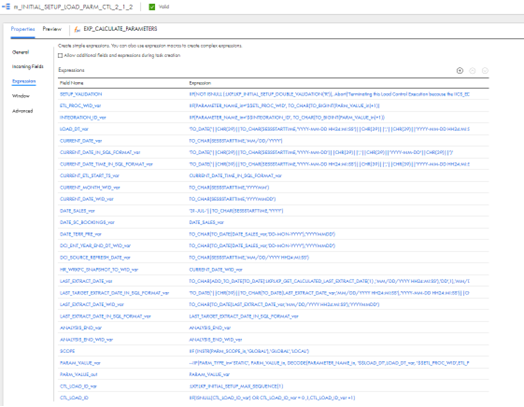
This task flow will update: Global parameter file -> Snowflake param table

This step will read the global parameters file with updated/calculated values and insert those values into the snowflake param table.

# Validation of Dynamic Parameter Values

All calculated parameters are calculated inside an asset ‘expression’ on mapping MTT\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2.

This mapping will read the param table on the snowflake, calculate necessary parameters, and load it into a global parameter file, with all updated parameters.



Each calculated parameter has its own rule of calculation, some have a sum of date, and the sum of value. Some of them are queries to be done on the snowflake.

The next page Screenshot shows the snowflake parameter table before the load control Process started

And parameters had their value before all the calculations happened in the load control Process.





Those screenshots, at the end of the page, show values after the load control process is finished and calculation mapping from load control ran and calculated all new values for all calculated parameters.





Those images show Only ‘calculated parameters’, as **I filtered for them on the query to print it**.

# The transition from task flow to task-flow (Control-M related)

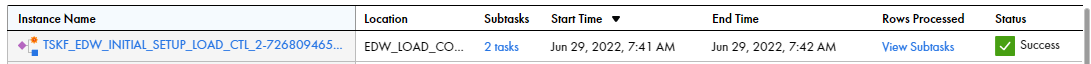
Control-M will kick off all taskflows, all 25 from EDW plus 2 (initial and end) from load control.

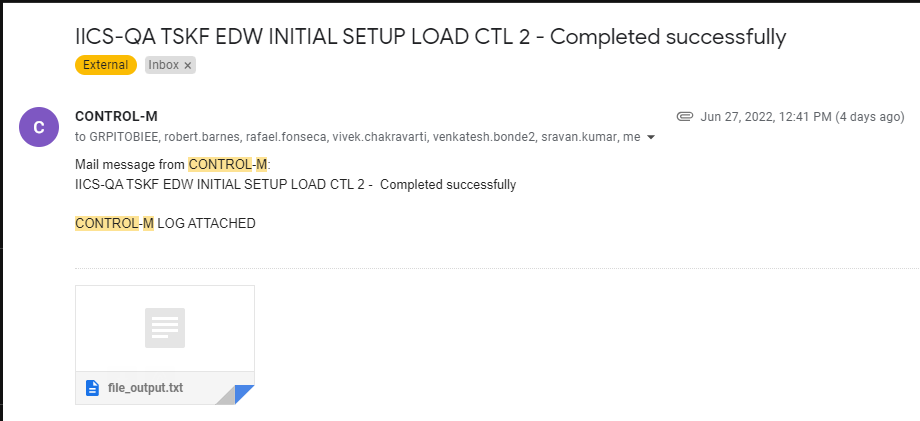
**\*\*\*screen shots from control-M side\*\*\***

When a taskflow is kicked off by Control-M, a command-line runs and the taskflow starts automatically on IICS.

If it completes successfully on IICS, an email from Control-M will be sent, accusing successful, with the runAJobCli log attached.

(IICS)



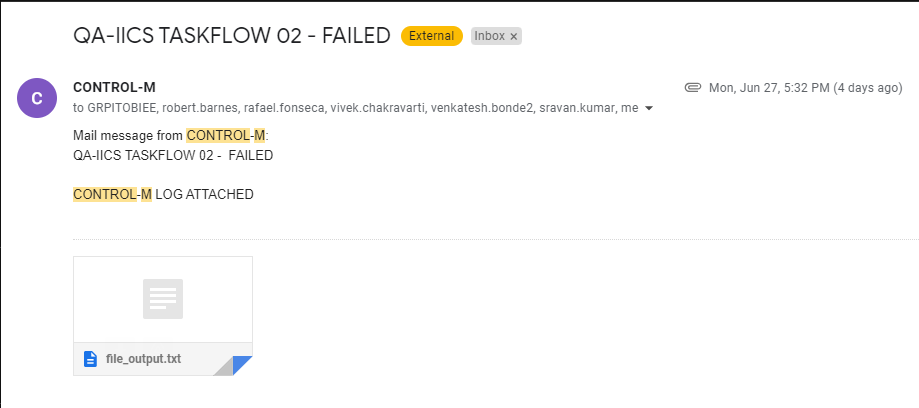
(CONTROL-M)  


If it gets suspended on IICS, an email from control-M accusing failure will be sent

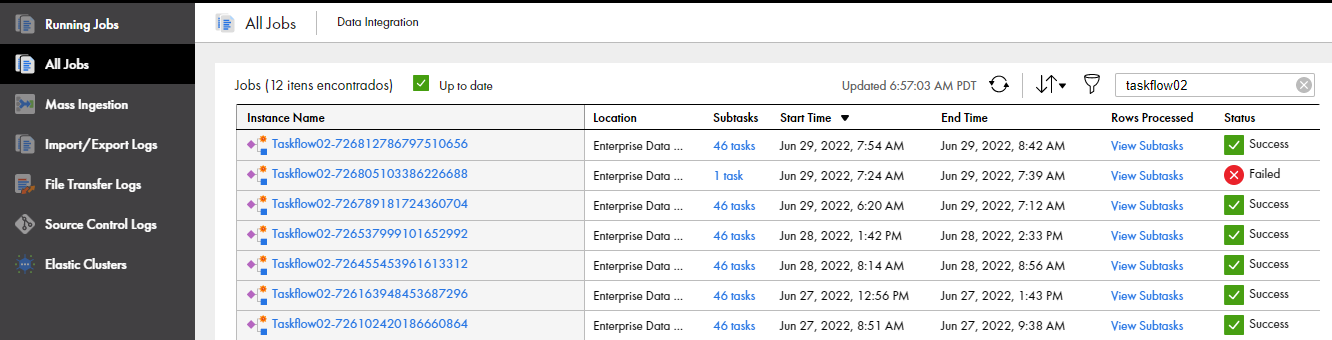
(IICS)



(CONTROL-M)



Once a task flow is started by Control-M, IICS users can monitor it on the ‘Monitor’ iics screen and follow it in real-time its execution.



# Load Control Taskflow Table on Snowflake

Every single taskflow of the EDW has a initial mapping task that will check if that taskflow is not currently running, or has already completed on that execution ID.

It checks on snowflake taskflow table if the taskflow has already a ‘R’ or ‘C’ register, for that CTL\_LOAD\_ID the is currently open.



The example on image above, taskflow02 is with C and CTL\_LOAD\_ID = 16, means it was already completed for the CTL\_LOAD\_ID 16.

Once a taskflow has not started for that CTL\_LOAD\_ID yet, it will have no register on the table for that exactly taskflow, then, the initial mapping task will allow it to continue, and insert new row on taskflow table, with a ‘R’ of running, for that opened CTL\_LOAD\_ID.

Every single taskflow of the EDW has a final mapping task that will update LOAD\_STATUS with a ‘C’ of completed on the taskflow that is being finished on that execution ID (CTL\_LOAD\_ID).



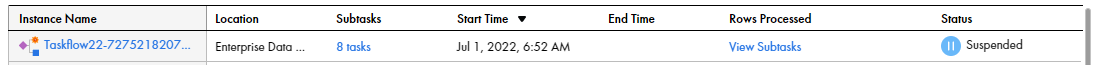
In short: First mapping of EDW Taskflows check if it not with a R (running) yet, or a C (already completed) and if not, starts the taskflow and set LOAD\_STATUS with a ‘R’ of running.

Last mapping will close the taskflow running with a C on the LOAD\_STATUS on the taskflow table.

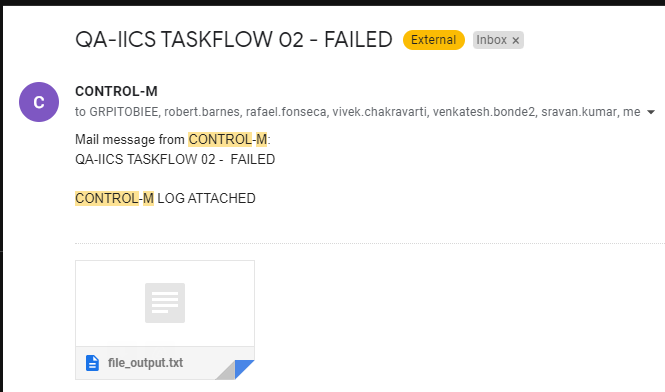
# Failure and Recovery Process

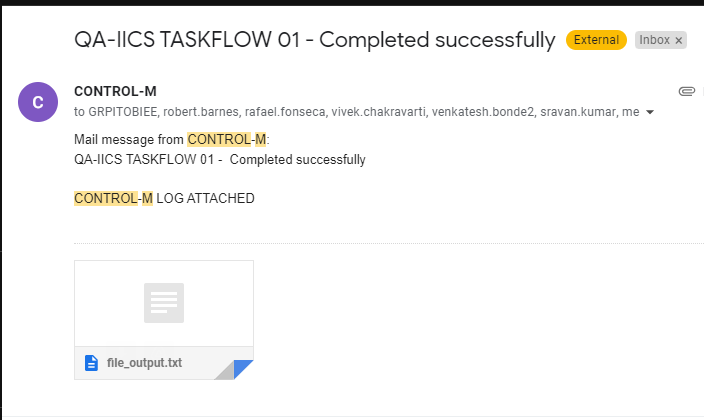
## Fail to Recovery

1. When any failure occurs. This Taskflow is going to get Suspended Status.



1. Control-M Job will get a failed status and send an email notification to the BI Team accusing failure on that taskflow.



1. The Suspended Taskflow will be Resumed by restApi by the Operator/BI Team (to confirm who). After fixes were done by the team.
2. The Operator/BI Team will Request Force-OK for the Control-M Job after the Taskflow is Successfully completed, so Control-M has successfully on that taskflow status.
3. Control-M will send an email notification to the BI Team accusing successful conclusion on that taskflow.  
     
   

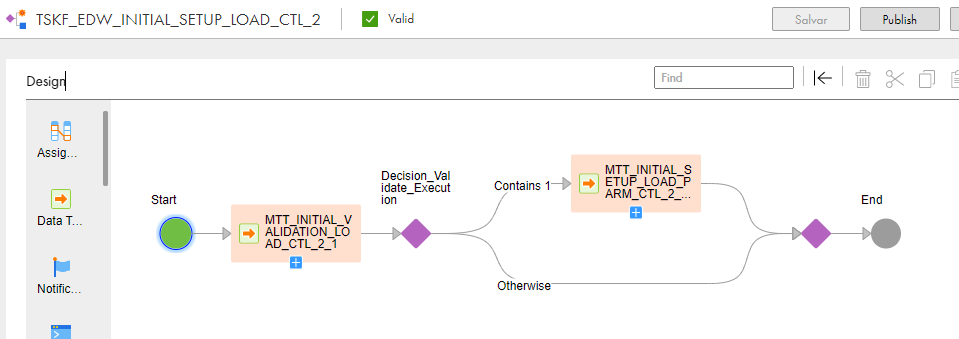
# ETL process setup (Load control + EDW Taskflows)

The whole setup has a total of 27 taskflows on it, meaning 2 taskflows from the load control process, and 25 from EDW load.

Below, there will be each one of them in order of execution and what they do regarding load control and parameters.

## Starting off:

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



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

It is the First Load Control Mapping Task that runs in the taskflow, and the first ever to run in the whole process. 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.

**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 and inserts a record in the Load Control Setup table in order to generate an Execution Plan.

This mapping will insert a new row on the Load Control Setup table and set LOAD\_STATUS field as ‘R’ of running.



Then, it comes the taskflow01, the first taskflow from EDW Load.

The INITIAL and END mapping task of EVERY EDW Load taskflow will be tasks from load control, and then in the Middle, you have the EDW mapping tasks. This will be standard for all the 25 EDW taskflows.

## First mapping of the taskflow01 (INITIAL MAPPING TASK):

**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 taskflow.

After that, you’ll have all the data tasks related to EDW mappings in the Middle of the taskflow.

And at the end, there is the END mapping task as the LAST one, from load control.

**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.

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

Receive the Name of the EDW Taskflow as input parameter (because there is no buit-in system variable to be used at Mapping level), and the mapping Will use the taskflow name as reference to update the taskflow table to LOAD\_STATUS = C of completed.

After explaining the INITIAL and END mappings of Tasfklow01, keep in mind that this structure Will be the same for the very next 24 taskflows, meaning: from 01 to 25 EDW Taskflows.

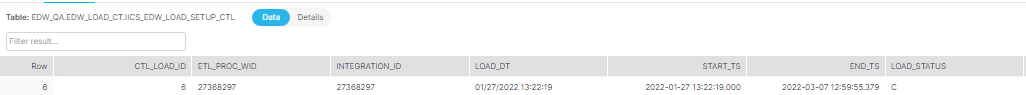
So we’ll be jumping to the END taksflow, considering 01 to 25 are already explained and Will be the same as taskflow01 above.

**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 (25) successfully completes. It represents the whole Step 5 in the Load Control Process.

This Taskflow is responsible to CLOSE the Execution Plan opened by the first mapping of the INITIAL taskflow (MTT\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2) indicating that its execution has successfully completed (updating LOAD STATUS AS ‘C’) and save the Calculated parameters from globalparameter file in the snowflake Parameter Table (IICS\_EDW\_PARAM\_CTL). It is a Sequencial Taskflow having two Mapping Tasks:

1. MTT\_END\_SETUP\_LOAD\_CTL\_5\_0 (this mapping closes the execution plan with C)



1. MTT\_END\_SETUP\_LOAD\_CTL\_UPDATE\_PARAM\_5\_0 (this mapping UPDATES parameter table from global parameter file on agent directory.)

