(IICS) Task-flow Orchestration Design Document

# Introduction

This document stands for an explanation of the entire Task-flow Code Design and workflow.   
  
In the next pages below, you will be able to learn and understand how the Donaldson IICS main and secondary task-flow orchestration was set and how this activates the Load Control process and update Snowflake Cloud Datawarehouse Tables and Objects.  
Passing through first-time Setup, daily process automation, important Task-Flows, Mapping-Tasks, and Mappings.

The whole point of this document is to instruct and make it possible for Donaldson IICS Users being able to reproduce this in all three environments (Development, Quality Assurance, and Production) and have the knowledge of how each field in the chain works and was developed by the team.  
  
In the first pages will be a one-time-only setup, then the rest of the pages might show the automation itself working daily.

# Index

[Introduction 1](#_Toc114828961)

[Index 2](#_Toc114828962)

[First Parameter File Generation (One time Only) 3](#_Toc114828963)

[Main Execution Plan Master Task-flow (One time Only) 6](#_Toc114828964)

[Multiple Region Parameter Configuration 13](#_Toc114828965)

[The Four Parameter Values for Each Seven Regions 15](#_Toc114828967)

[Load Control Process Validation (Parameter Validation Check) 18](#_Toc114828968)

[Parameter Validation Check (static values code) 19](#_Toc114828969)

[Adding Additional Parameter 22](#_Toc114828970)

[Main Execution Task-Flow’s Plan Orchestration (01…25) 25](#_Toc114828971)

[Secondary Execution Plan (Independent Running) 27](#_Toc114828972)

[Email Notifications 29](#_Toc114828973)





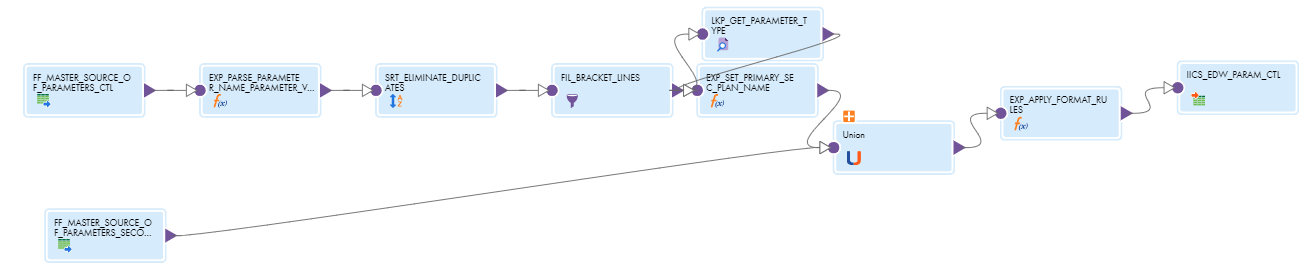
# First Parameter File Generation (One time Only)

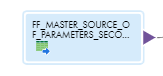
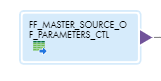
**First Time Setup**

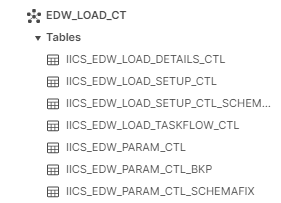
The first step is the Generation of the right parameter filename at the right location for the main and secondary execution plans.  
The first time when all the EDW Loads will start from IICS (not from PowerCenter anymore), it is important to load the last parameter file generated in PowerCenter so the 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

**Parameter Load Mapping:**

**m\_LOAD\_PC\_DAC\_PARAMETERS\_TO\_IICS\_SNOWFLAKE\_PARAMETERS**



Where the sources in the Mapping are both an ETL Union containing**** **FF\_MASTER\_SOURCE\_OF\_PARAMETERS\_CTL** Source attached to its Sorters, Filters, Lookups, and Expressions. And the other Source flat file with the name: **FF\_MASTER\_SOURCE\_OF\_PARAMETERS\_SECONDARY**This mapping is a one-time load mapping therefore there is no Mapping Task/Task**-**flow Associated with it. This mapping truncates the target (parameter table in Snowflake), so when running this mapping it is recommended to be careful.

Logged In Snowflake you will find in the path:  
Database: EDW\_DEV  
Schema: EDW\_LOAD\_CT  
and  
Table-Name: **IICS\_DW\_PARAM\_CTL** 

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

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

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

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

Run the Mapping

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

**First 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 the snowflake setup table (CTL\_LOAD\_ID).

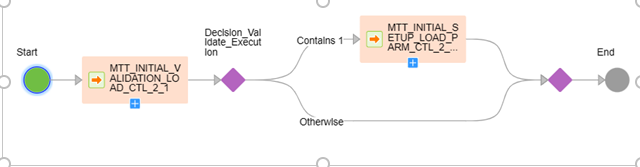
It is the First Load Control Task-flow that is invoked by Control-M before any EDW Load starts. It represents the whole part of eliminating duplicates in the Load Control Process.

This Task-flow 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.

**Initial Setup to Load Main Task-flow’s**

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’**

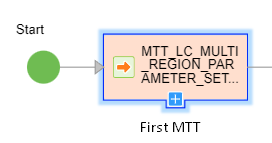
This Task-flow will update: Global parameter file -> Snowflake param table

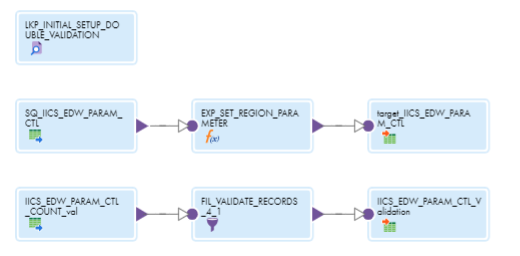
This step will read the global parameters file with updated/calculated values and insert those values into the snowflake param table.

THIS WILL BE ONLY RUN-ON IN THE MIGRATION DAY   
For more information please check [Import/Export](https://donaldson.sharepoint.com/sites/IT-Project-Mgmt-Office-Enterprise-Data-Platform/Shared%20Documents/Technical%20Documents/Orchestration%20and%20Scheduling/IICS%20Import&Export%20Deployment%20Playbook.docx) and [Control-M x IICS](https://donaldson.sharepoint.com/sites/IT-Project-Mgmt-Office-Enterprise-Data-Platform/Shared%20Documents/Technical%20Documents/Orchestration%20and%20Scheduling/IICS%20Control-M%20Integration%20Playbook.docx)   
----------------------------------------------------------------------------------------------------------------------------------

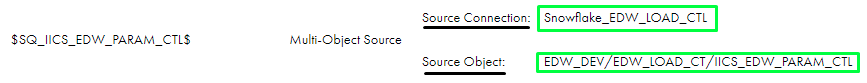
# Main Execution Plan Master Task-flow (One time Only)

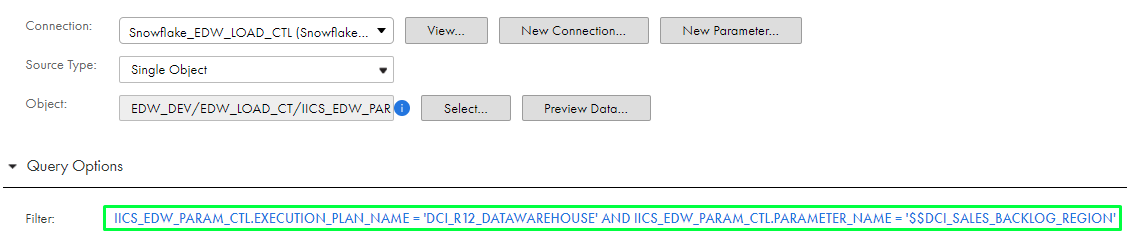
**Main Execution Plan Design**

The main execution plan will act in the **‘DCI\_R12\_DATAWAREHOUSE’** Task-flow.  
  
On the other hand, the Mapping Tasks contained in the **‘DCI\_R12\_DATAWAREHOUSE’** Task-flow will work as a sequence. The first mapping-task of it is **‘MTT\_LC\_MULTI\_REGION\_PARAMETER\_SETUP’**  
  
and this **“MTT”** might call its containing map **‘m\_LC\_MULTI\_REGION\_PARAMETER\_SETUP’.**  

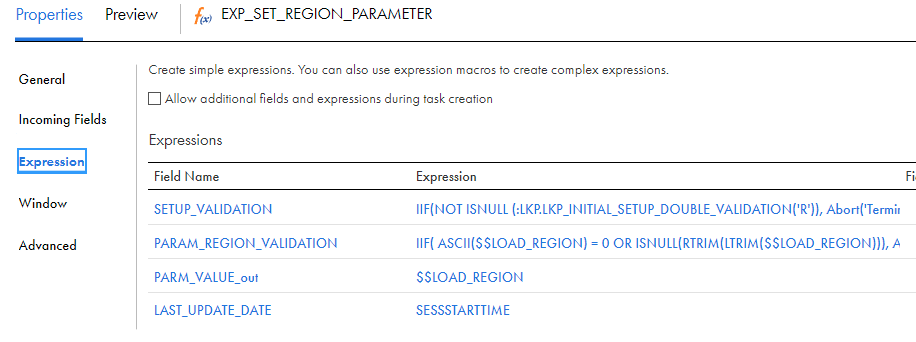
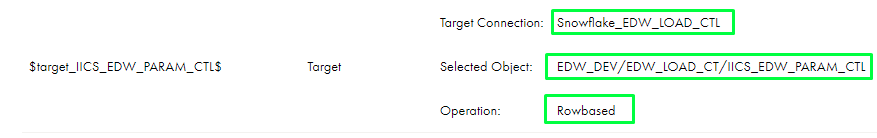

This map is composed of two ‘**Source’s**’, one ‘**Expression**’ function, two ‘**Target’s**’, one ‘**Filter**” function, and a ‘**Lookup**’.  


* The first **Source** ‘**SQ\_IICS\_EDW\_PARAM\_CTL**’:  
  

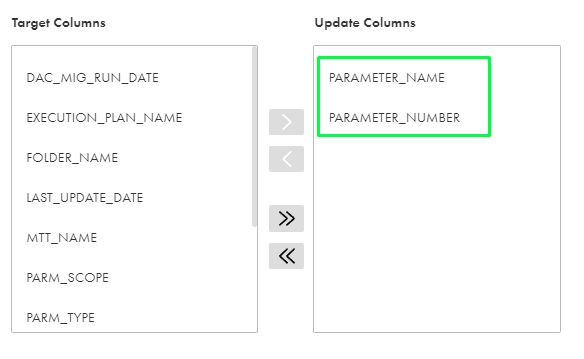
This object on the map will be a query in the Snowflake Cloud Data-Warehouse Table, in this case, the development (DEV) environment:  




This Source filter (**as a Query**) the table **‘IICS\_EDW\_PARAM\_CTL’** with the condition: **\*IICS\_EDW\_PARAM\_CTL.EXECUTION\_PLAN\_NAME = 'DCI\_R12\_DATAWAREHOUSE' AND IICS\_EDW\_PARAM\_CTL.PARAMETER\_NAME = '$$DCI\_SALES\_BACKLOG\_REGION'\*** and this query states to bring up **ONLY** the Region-Parameter contained in the ‘**Main-Task-flow’s’** (Referring to 1…25 Task-Flow’s).

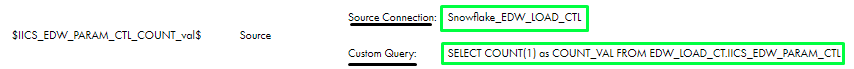
Connected to the first Source comes the ‘**Expression**’ function,  
**’EXP\_SET\_REGION\_PARAMETER’**:  
  
This expression codifies if the Region Execution Plan Parameter (The whole Task-flow) **“was, is being, or will be”** executed.   
Evidence of **Expression** field codes mentioned before below:  
  
The first Target is **‘target\_IICS\_EDW\_PARAM\_CTL’**.  


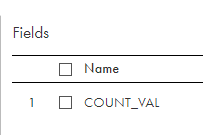
This target is composed by a **‘Data-Driven’** operation that fills the **“EDW\_PARAM\_CTL”** Snowflake Cloud table with the previous expression row’s updating **“PARAMETER\_NAME”** and **“PARAMETER\_NUMBE”** columns. Graphical user interface, text

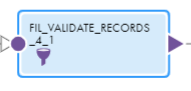
Description automatically generated

* The second **‘Source’**

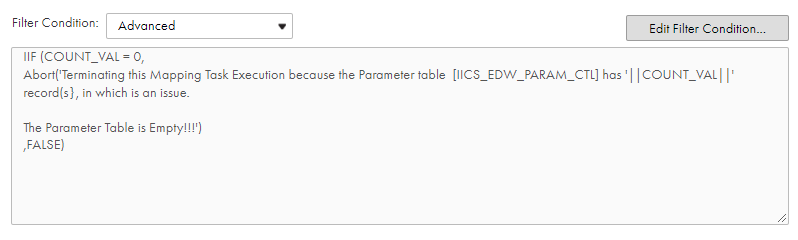
Stands for a **SELECT** row **COUNT\_VAL** in the same **‘EDW\_PARAM\_CTL’** Snowflake table used as target in the previous target explanation. This query starts the validation of the new **rows** in CTL table.

  
The filter is set directly in **Query Mode** shown below:

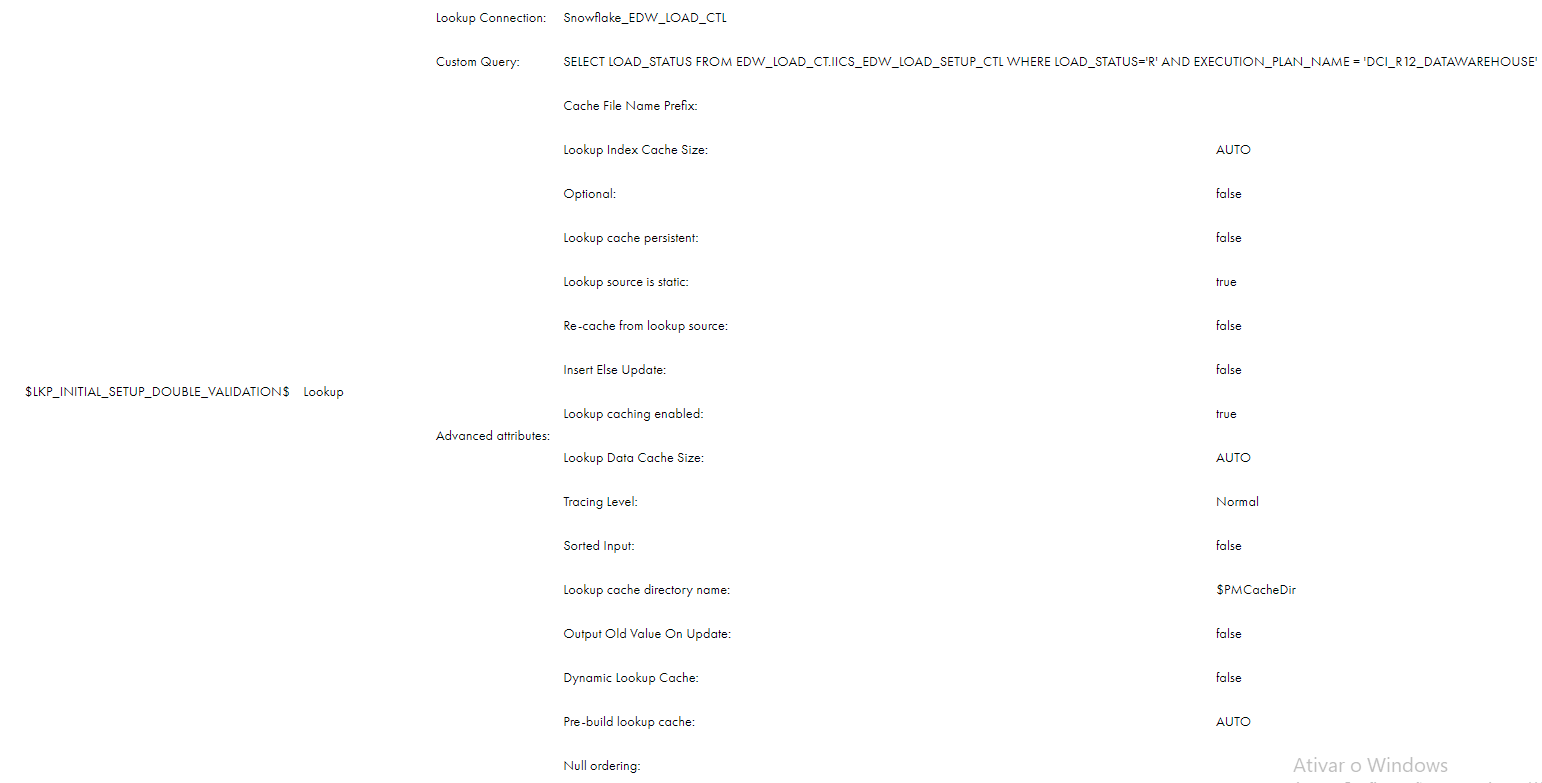
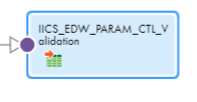
Bringing only the **COUNT\_VAL** as field:  


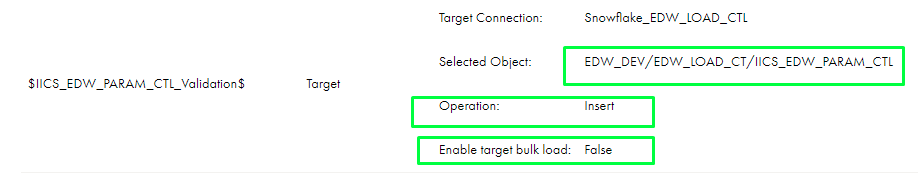
Attached to previous Source comes a **‘Filter’** transformation that wil fail the mapping if:

The number of records from the Source(EDW\_PARAM\_CTL) that is different than 1.  
  
Otherwise, it will just filter out using **FALSE statement** as the records from the Source pipeline will not need to go to the target.

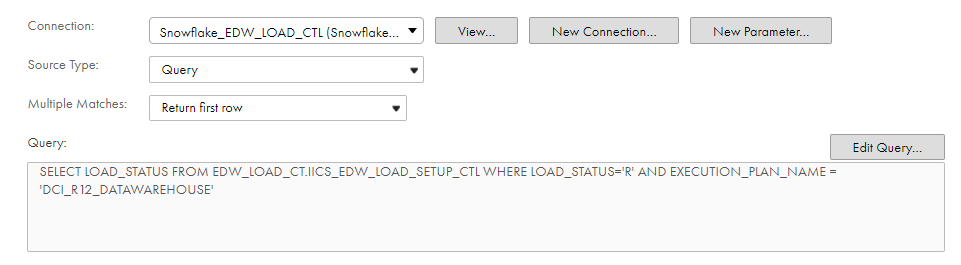


* The Second **‘Target’**:

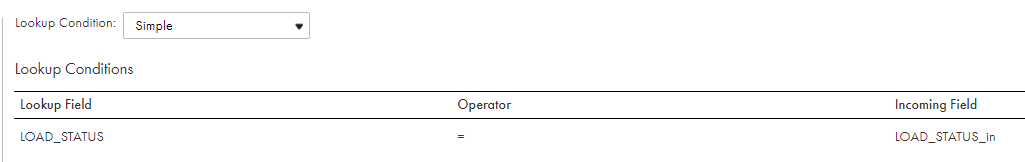
Following the sequence after the Filter comes the ‘Target’ validation table operation.  


This will validade the previous filter in the previous transformation and applying an **Insert operation** of the values in the ‘IICS\_EDW\_PARAM\_CTL’.

Last passive transformation in the Mapping is a passive **Lookup** transformation that queries “**SELECT LOAD\_STATUS FROM EDW\_LOAD\_CT.IICS\_EDW\_LOAD\_SETUP\_CTL WHERE LOAD\_STATUS='R' AND EXECUTION\_PLAN\_NAME = 'DCI\_R12\_DATAWAREHOUSE’**” similar to the first Source on the mapping but this time searching if any Execution Plan is Running.



This means:   
  
Has the ‘**R**’ value on the column **LOAD\_STATUS** and if it matches with **the EXECUTION\_PLAN\_NAME – DCI\_R12\_DATAWAREHOUSE** in load control table.  
It verify if there is already a execution Running in the **IICS\_EDW\_LOAD\_SETUP\_CTL.**



The **Main Execution Plan** is set up by a **Mapping Task that alters the registers in the Snowflake Param-Table on the Column LOAD\_STATUS and inserts them into ‘R’ which stands for ‘RUNNING’ on each.** The Task-flow reports itself that it is on a **‘RUN’** once its time comes.  
**TSKF\_EDW\_INITIAL\_SETUP\_LOAD\_CTL\_2**In the sequence, this mapping task opens the table registering in Snowflake Param Table.

Then the Task-flow **TSKF\_EDW\_END\_LOAD\_CTL\_5** starts his job that is:  
When each Task-flow is completed.  
**All of them finish the Task-flow and changed their register to ‘C’ meaning ‘COMPLETED’ in the Snowflake Param-Table**.

The Primary Execution plan known as Main Execution Plan is the 1 to 25 “Task-Flows” will be running in between **TSKF\_EDW\_INITIAL\_SETUP\_LOAD\_CTL\_2 and TSKF\_EDW\_INITIAL\_SETUP\_LOAD\_CTL\_2.**

# Multiple Region Parameter Configuration

**Multiple Regions First Setup**

* The five Region Parameter Setup at this point, will call all main Task-flows to run.
* They need to run for each specific region and to this:

Must create a specific command-line to each region separated.

* Those commands are being used in Control-M and the scheduling itself might be done there. Check [Control-M Jobs Setup and Frequency](https://donaldson.sharepoint.com/:x:/r/sites/IT-Project-Mgmt-Office-Enterprise-Data-Platform/Shared%20Documents/Technical%20Documents%20(David)/IICS%20Technical%20Documents/Control-M%20Jobs%20Setup%20and%20Frequency.xlsx?d=w529148f06a37486c8c03e3b58a619063&csf=1&web=1&e=Hlsq3V)

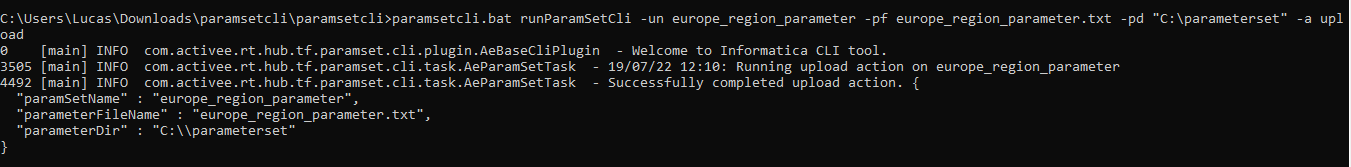
**Creating Each Region Parameter**

The process of creating Region Parameter is the same as creating any other parameter text-file [explained before](#_Parameter_Validation_Check) and putting static values in it.

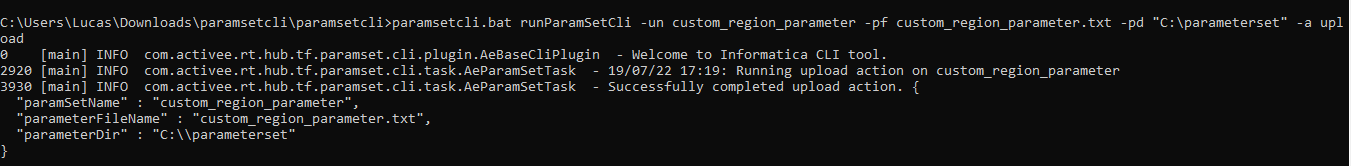
In the next steps user might be able to see the **Region-Parameter** the team has already created.

Keep in mind:  
[Import/Export](https://donaldson.sharepoint.com/sites/IT-Project-Mgmt-Office-Enterprise-Data-Platform/Shared%20Documents/Technical%20Documents/Orchestration%20and%20Scheduling/IICS%20Import&Export%20Deployment%20Playbook.docx) Documentation from DEV to QA or PRD, [Control-M x IICS](https://donaldson.sharepoint.com/sites/IT-Project-Mgmt-Office-Enterprise-Data-Platform/Shared%20Documents/Technical%20Documents/Orchestration%20and%20Scheduling/IICS%20Control-M%20Integration%20Playbook.docx) management and [Scheduling](https://donaldson.sharepoint.com/:x:/r/sites/IT-Project-Mgmt-Office-Enterprise-Data-Platform/Shared%20Documents/Technical%20Documents%20(David)/IICS%20Technical%20Documents/Control-M%20Jobs%20Setup%20and%20Frequency.xlsx?d=w529148f06a37486c8c03e3b58a619063&csf=1&web=1&e=Hlsq3V) may be found in linked documentation.

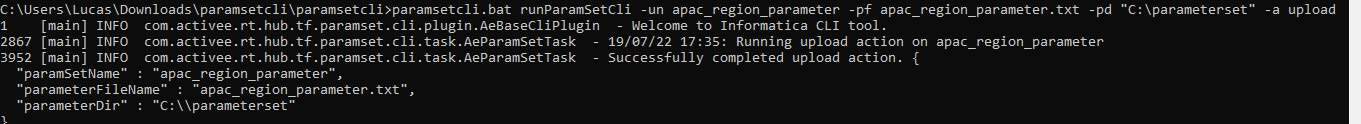
* **Europe Region Parameter**  
  Use:   
  paramsetcli.bat runParamSetCli -un europe\_region\_parameter -pf europe\_region\_parameter.txt -pd "C:\parameterset" -a upload



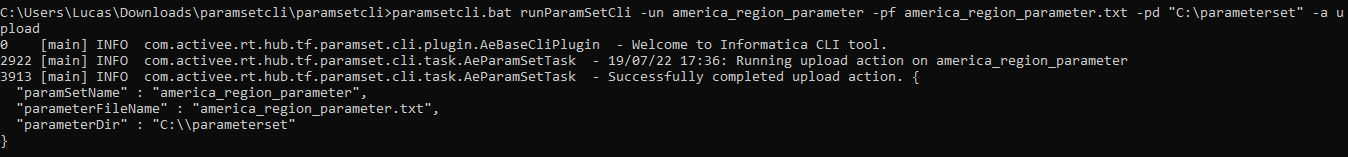
* **Custom Region Parameter**Use:  
  paramsetcli.bat runParamSetCli -un custom\_region\_parameter -pf custom\_region\_parameter.txt -pd "C:\parameterset" -a upload



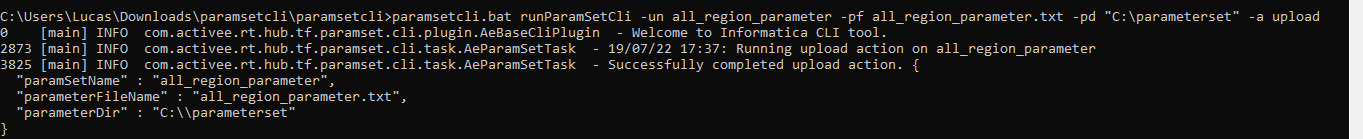
* **Asia Pacific Region Parameter**  
  Use:paramsetcli.bat runParamSetCli -un apac\_region\_parameter -pf apac\_region\_parameter.txt -pd "C:\parameterset" -a upload



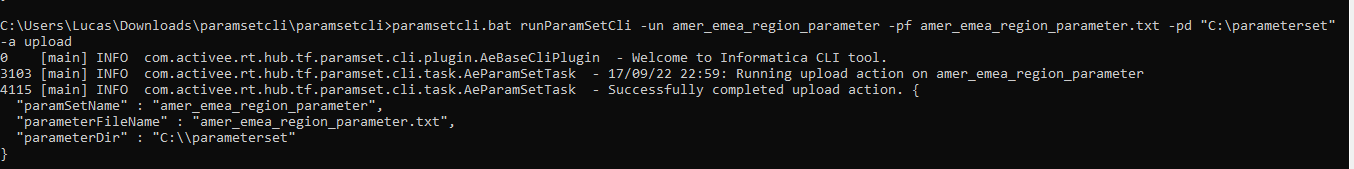
* **America Region Parameter**  
  Use:  
  paramsetcli.bat runParamSetCli -un america\_region\_parameter -pf america\_region\_parameter.txt -pd "C:\parameterset" -a upload



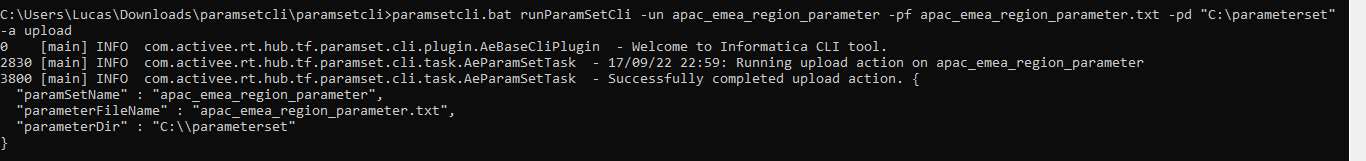
* **All Region Parameter**  
  Use:  
  paramsetcli.bat runParamSetCli -un all\_region\_parameter -pf all\_region\_parameter.txt -pd "C:\parameterset" -a upload



* **North, Central, and South America / Europe, the Middle East, and Africa Region Parameter**  
  Use:  
  paramsetcli.bat runParamSetCli -un amer\_emea\_region\_parameter -pf amer\_emea\_region\_parameter.txt -pd "C:\parameterset" -a upload

****

* **Asia Pacific and Africa / Europe, the Middle East, and Africa Region Parameter**  
  Use:  
  paramsetcli.bat runParamSetCli -un apac\_emea\_region\_parameter -pf apac\_emea\_region\_parameter.txt -pd "C:\parameterset" -a upload



The Four Parameter Values for Each Seven Regions  
  
Each region will receive 4 parameters values related to their own Region Setup.  
  
**[GLOBAL]  
$$LOAD\_REGION=** VALUE **$$DCI\_TZ\_GMT\_OFFSET\_FROM=** VALUE **$$ DCI\_TZ\_GMT\_OFFSET\_TO=** VALUE

**$$ DCI\_TZ\_GMT\_INV\_DAY\_ADJ=** VALUE

All Region Parameter set  
[GLOBAL]  
$$LOAD\_REGION=’DON US BG’, ‘DON MX BG’, ‘DON AP BG’, ‘DON EU BG’  
$$DCI\_TZ\_GMT\_OFFSET\_FROM=-12  
$$ DCI\_TZ\_GMT\_OFFSET\_TO=14

$$ DCI\_TZ\_GMT\_INV\_DAY\_ADJ=’DAY\_AGO’America Region Parameter set  
[GLOBAL]  
$$LOAD\_REGION=’DON US BG’, ‘DON MX BG’   
$$DCI\_TZ\_GMT\_OFFSET\_FROM=-12  
$$ DCI\_TZ\_GMT\_OFFSET\_TO=-2

$$ DCI\_TZ\_GMT\_INV\_DAY\_ADJ=’DAY\_AGO’  
  
Asian Pacific Region Parameter set  
[GLOBAL]  
$$LOAD\_REGION= ‘DON AP BG’   
$$DCI\_TZ\_GMT\_OFFSET\_FROM=5  
$$ DCI\_TZ\_GMT\_OFFSET\_TO=14

$$ DCI\_TZ\_GMT\_INV\_DAY\_ADJ=’CURRENT\_DAY’

Europe Region Parameter set  
[GLOBAL]  
$$LOAD\_REGION=‘DON EU BG’  
$$DCI\_TZ\_GMT\_OFFSET\_FROM=-1  
$$ DCI\_TZ\_GMT\_OFFSET\_TO=4

$$ DCI\_TZ\_GMT\_INV\_DAY\_ADJ=’CURRENT\_DAY’

**The Region Parameter Mapping**



Will receive 4 In Out listed below:  
Graphical user interface, text

Description automatically generated with medium confidence

The mapping **‘Source’** will use the previous parameters to query in the Snowflake PARAM\_CTL table.

Text, Word

Description automatically generated

Text

Description automatically generated with medium confidence

Text

Description automatically generated

Then the **‘Expression’** contained in the mapping will decode the parameters function.  
Text

Description automatically generatedText

Description automatically generated with medium confidence

Listing the previous values of each parameter Input:Text, letter

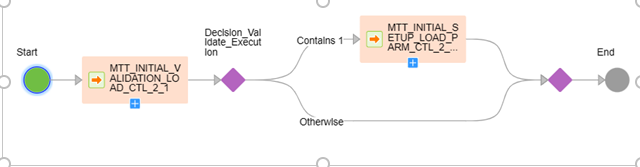
Description automatically generated

# Load Control Process Validation (Parameter Validation Check)

**Load Control Daily Run**

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’ job:**

* This Task-flow will update: Global Parameter File -> Snowflake Param Table
* At this step.  
  The Task flow will read the **global parameters** file with **updated/calculated** values and insert those values into the **snowflake param table**.
* The Main Execution Plan is set up by a Mapping Task that alters the registers in the Snowflake Param-Table on the Column **LOAD\_STATUS** and inserts them into **‘R’** which stands for **‘RUNNING’** on each. The Task-flow reports itself that it is on a ‘RUN’ once its time comes.  
  **TSKF\_EDW\_INITIAL\_SETUP\_LOAD\_CTL\_2.**
* In the sequence, this mapping task opens the table registering in Snowflake Param Table.
* Then the Task-flow **TSKF\_EDW\_END\_LOAD\_CTL\_5** starts his job that is:  
  When each Task-flow is completed.  
  All of them finish the Task-flow and changed their register to **‘C’** meaning **‘COMPLETED’** in the Snowflake Param-Table.
* The Primary Execution plan known as Main Execution Plan is the 1 to 25 “Task-Flows” will be running in between the **TSKF\_EDW\_INITIAL\_SETUP\_LOAD\_CTL\_2** process.

# Parameter Validation Check (static values code)

**The Parameters Mapping Workflow (In the Load Control Task-flow)**

* To the process of load control start. First it validates the parameter values that are dynamic parameters based in the logic in mapping **‘MTT\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2’,** where all calculated parameters are described and receive their relations.
* They work through a function inside the **‘Expression’** on the mapping ‘**m\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2’** that will always be called once any Task-flow in this case the **Load-Control Task-flow** that contains the **Mapping Task** **‘MTT\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2’**.
* The Mapping **‘m\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2’** code all the load control important fields.  
   
* The Mapping Expression fields:  
  **Text

  Description automatically generated**Text

  Description automatically generated with medium confidence

Table, timeline

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Therefore, all the **“Lookups**” in the mapping must validate the existence and status of each described field in the **“Expression”** also in the same mapping.Text

Description automatically generated with medium confidence

Sample of Column Field used to create and transform a Value into a Variable to “Extract Last Target Date in SQL Format” developed.

A picture containing graphical user interface

Description automatically generated

In the Load Control Task-Flow process:  
The mapping-task **MTT\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2** will read the param table on the snowflake, calculate necessary parameters, and load it into a global parameter file, with all updated parameters.

Graphical user interface, text, application

Description automatically generated

# Adding Additional Parameter

**Parameter Addition Setup**

To add new parameters in IICS first. It’s needed to first to setup any machine with these configurations:

* Download and Install Java SE Runtime Environment 8 or newer. That can be found in this link clicking [here](https://www.oracle.com/br/java/technologies/javase/javase8-archive-downloads.html).  
  Graphical user interface

  Description automatically generated

Once Java 8 is downloaded and installed you can continue with next steps.

* Download and unzip this file:  
  **“paramsetcli”**.  
  That you will find by clicking [here](https://knowledge.informatica.com/s/article/DOC-19232?language=en_US). An Informatica website.  
  Graphical user interface, text, application

  Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

* Once extracted.  
  The **first folder** should look like this:  
  Logo

  Description automatically generated

And inside it might be another **“paramsetcli”** like this:

A picture containing graphical user interface

Description automatically generated Graphical user interface, text, application

Description automatically generated

Then in the next step, it’s important to have the Donaldson IICS Org Information to continue.  
Then open **restenv.properties** as a text-file and write down Org Information to fill this document.

That can be opened with any text editor such as:  
Notepad++(Windows/Mac),

Nano (Linux),

Gedit (Linux),

Notepad (Win) and etc…

The information you will need to fill will come like this:  
Graphical user interface, text

Description automatically generated

And should be completed like this:  
Graphical user interface, text, application

Description automatically generated

Username and Password needs to be exactly the same as the credentials the user utilize in Browser IICS.

Remember “**PROXYHOST=”** and “**RETRYCOUNT=”** Doesn’t need to be filled

Contact the IICS Administrator to provide it for you.

**Creating and Adding New Parameters (Parameter / Value)**

With all set.

Now you can start adding new text parameters.

For example:

* Let’s suppose the user wants to add the **“europe\_region\_parameter”.**

You must start a new text file, to the standard pattern it will be used: **“europe\_region\_parameter” as Parameter Name** for this to be recognized and called in IICS workflows (T.F.s/M.T.T.s/M.s).

* This is the **Parameter Name** that the IICS Organization Repository directory (inaccessible) will store the text file in cloud.
* Next step is creating the text-file that again should be patterned as standards, in this case **“europe\_region\_parameter.txt”.**



Containing parameter **values** inside it.

Graphical user interface, text, application, table, email

Description automatically generated

* To upload those values in the text-file to IICS Repository Directory.  
  The user must open anything like a **Command Prompt** **(WinSCP/Putty/Linux Prompt)** and find the extracted folder **“paramsetcli”** in this case: “C:\Users\david\Desktop\paramsetcli\paramsetcli” so:  
  **cd C:\Users\david\Desktop\paramsetcli\paramsetcli**  
  A picture containing graphical user interface

  Description automatically generated
* Once the extracted directory was placed in prompt you must place this command bellow containing:

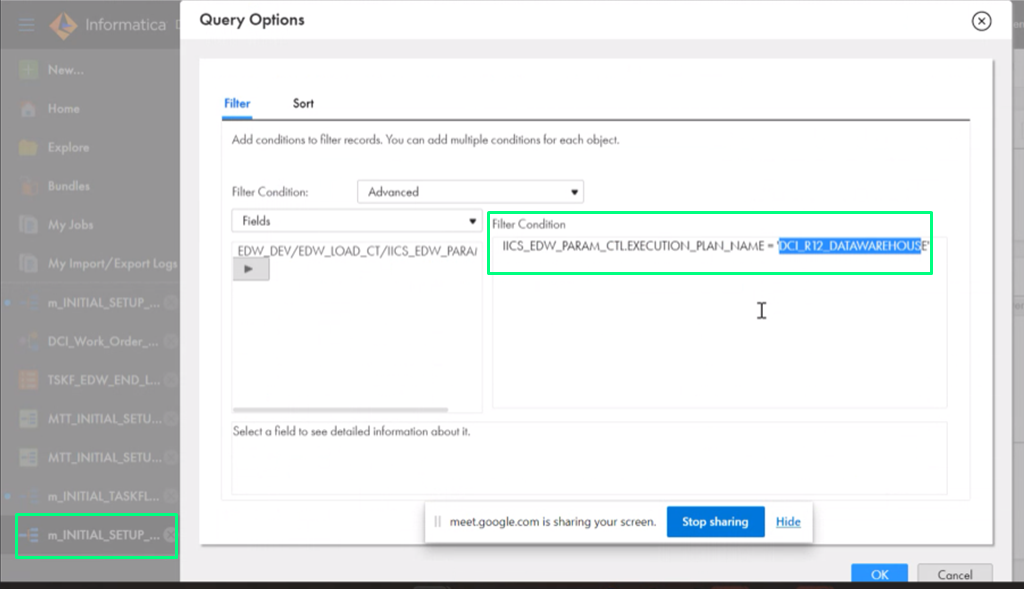
1. The current **parameter-name “-un”** (europe\_region\_parameter).
2. The **text-file name “-pf”** (europe\_region\_parameter.txt).
3. And the **file-path -pd** (C:\Users\david\Desktop\)  
   that contains it between **“”**:

* In this case:

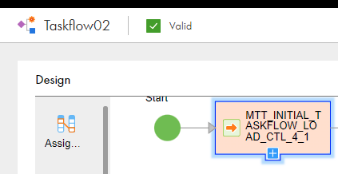
**paramsetcli.bat runParamSetCli -un europe\_region\_parameter -pf europe\_region\_parameter.txt -pd "C:\Users\david\Desktop\" -a upload**

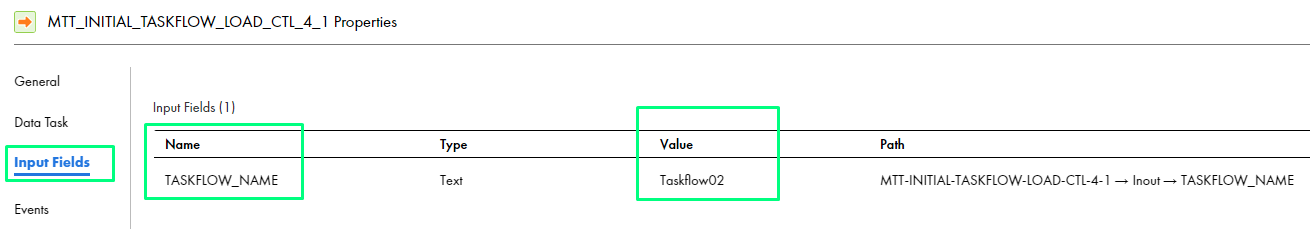
# Main Execution Task-Flow’s Plan Orchestration (01…25)

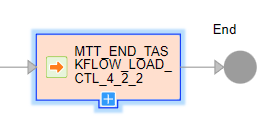
**Main Task-Flow’s Load Control Workflow Orchestration**

This Loads will be orchestrated by the validation of the Lookup’s that can be encountered in the Initial Mapping-Task from all Main Task-flow’s (01…25). 

* Those Main Task-flows (01...25) might be updating Register from ‘R’ meaning ‘RUNNING’ to ‘C’ meaning ‘COMPLETE’ in the Task-Flow Table on Snowflake Specific Task-Flow Table while being called and checked for run in the Snowflake Param-Table this task will be occurring when each pass threw his first and last Mapping-Task so:  
  ‘MTT\_INITIAL\_TASKFLOW\_LOAD\_CTL\_4\_1‘ and ‘MTT\_END\_TASKFLOW\_LOAD\_CTL\_4\_2\_2’ respectively.

The first Mapping-Task from each Main Task-flow will be addressing the name of it as a Value for an Input Field for example in “Taskflow02” Initial Mapping we have this (TASKFLOW\_NAME):

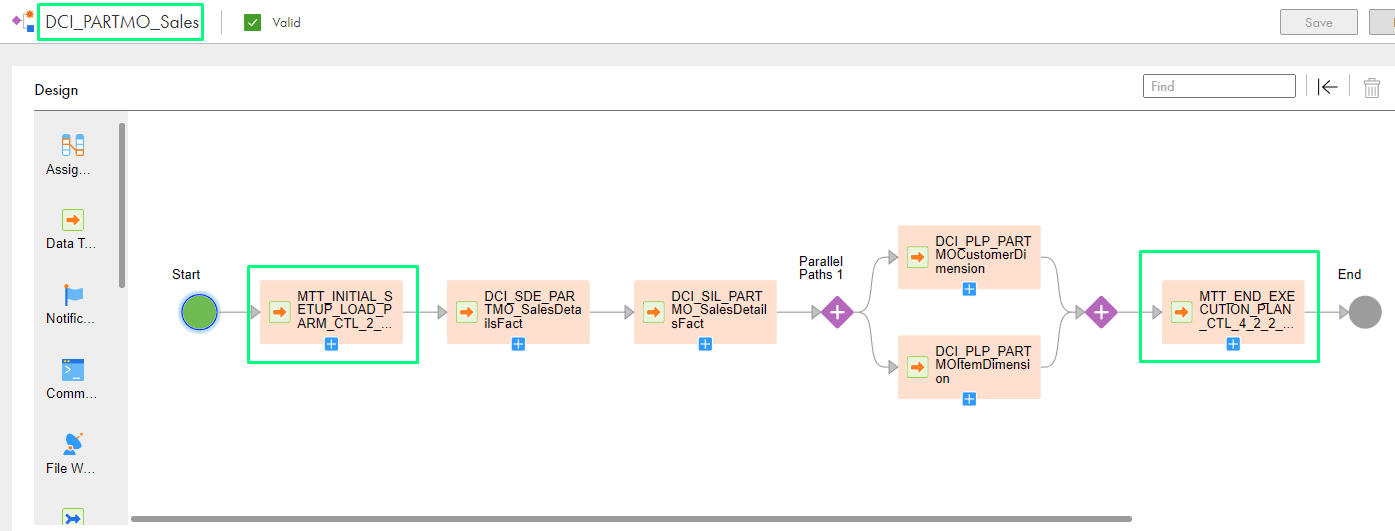
**‘MTT\_INITIAL\_TASKFLOW\_LOAD\_CTL\_4\_1’**  


* On the Main Execution Plan Task Flows, the Last Mapping Task from each one (‘MTT\_END\_TASKFLOW\_LOAD\_CTL\_4\_2\_2’) changes the register on Snowflake Taskflow-Table caring on again the Imputed Field (TASKFLOW\_NAME) and finishing the workflow until all the Main Task-Flows finishes. 

# Secondary Execution Plan (Independent Running)

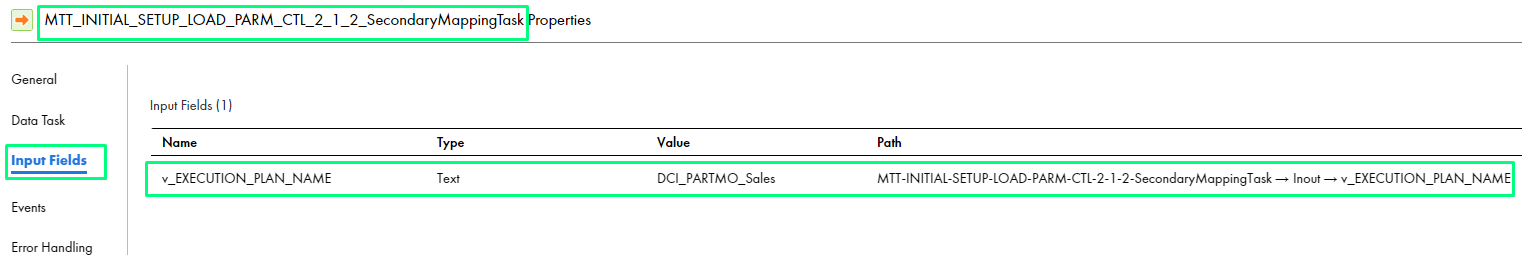
**Secondary Execution Plan Setup**

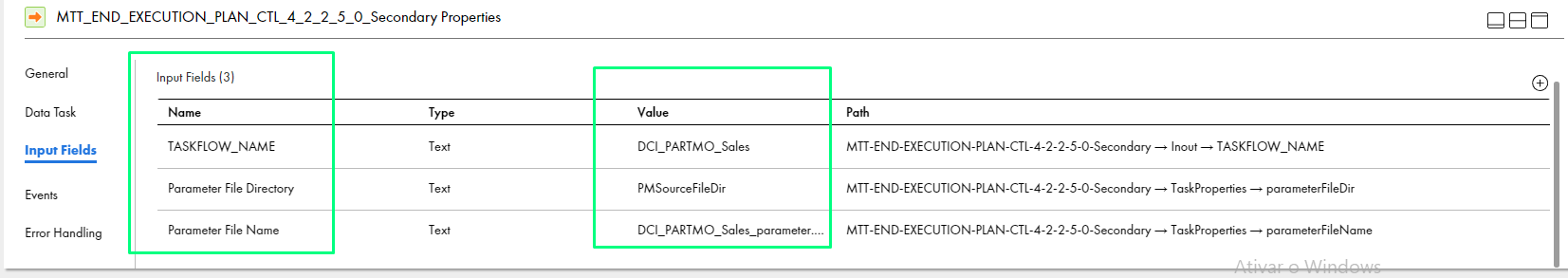
For Secondary Execution Plans:

* The workflow look’s the same as de Main Execution Plans but the difference between than is that the Secondary Task-flows has a different Initial and Final Mapping-Task Name.   
  Although they have the same design of Main Task-flows the Execution Plan show up on their initial Mapping-Tasks as an Input Field.
* Their **First** and **Last** Mappings are:  
  **‘MTT\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2\_SecondaryMappingTask’** 

**and**   
**‘MTT\_END\_EXECUTION\_PLAN\_CTL\_4\_2\_2\_5\_0\_Secondary’.**

* Those Mapping-Tasks have different In and Out Outputs Fields.
* Taking the First Mapping Task from the **‘DCI\_PARTMO\_Sales’** Task-flow you might be able to see the first Input Field Difference. The variable **‘v\_EXECUTION\_PLAN\_NAME’** becomes Input for the Secondary Task-Flow name for example **“DCI\_PARTMO\_Sales”**:  
  **‘MTT\_INITIAL\_SETUP\_LOAD\_PARM\_CTL\_2\_1\_2\_SecondaryMappingTask**’



* All the Finishing (End) Secondary Task-Flow Mapping-Task’s will have a difference too.  
  They all end with: **‘MTT\_END\_EXECUTION\_PLAN\_CTL\_4\_2\_2\_5\_0\_Secondary’**  
  But this time **‘TASKFLOW\_NAME’**, **‘Parameter File Directory’** and **‘Parameter File Name’** are set as Input Fields.  
  
* Their Parameter-File-Name becomes **‘Task-FlowName\_parameter.txt’** in this case: **DCI\_PARTMO\_Sales\_parameter.txt**
* Make sure that the Load ID and Task flow Name are in the Parameter file.

# Long Running Task flows (LR)

Graphical user interface, application

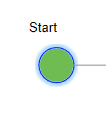
Description automatically generated with medium confidence

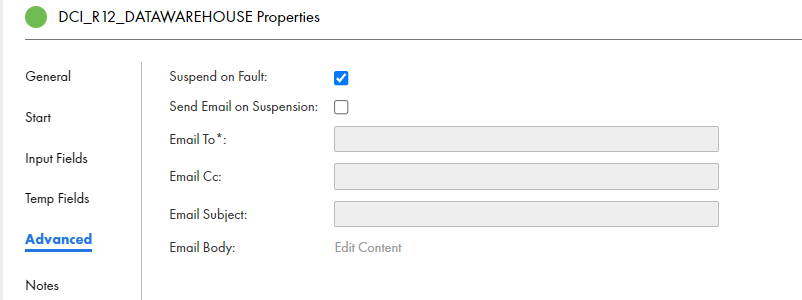
# Email Notifications

**Control-M mailing**

All IICS running jobs updates will be sent by Control-M.  
Those pieces of evidence can be found in the section Email from Control-M at 13 of the [IICS Control-M Integration Playbook](https://donaldson.sharepoint.com/sites/IT-Project-Mgmt-Office-Enterprise-Data-Platform/Shared%20Documents/Technical%20Documents/Orchestration%20and%20Scheduling/IICS%20Control-M%20Integration%20Playbook.docx).

**IICS Mailing**

* The IICS mailing must be configured in each IICS Task-Flow object.
* For “DCI\_R12\_DATAWAREHOUSE”, the main task-flow.
* Users can configure it by clicking in Task-flow Object.
* Hitting the **‘Start’** icon. 
* Then going to the section **‘Advanced’** in Task-Flow Properties.

Exactly like the image bellow:

* There you can apply the preferred Task-Flow mailing content.
* Such as:

1. Send e-mail on suspension
2. Email to specific contact
3. Copy other contacts
4. Write mail Subject
5. Write mail content

Mailing Setup DEV at DCI\_R12\_DATAWAREHOUSE taskflow:  
Graphical user interface, text, application, email

Description automatically generated

Sample of mailing in a DEV failure on Task-flow18:

Graphical user interface, text, application, email

Description automatically generated