# Wave



## Description

Wave is a external tool that calculate cost of saving on each project. Talend can get the data via API of this server and the output of the API is excel file.

It is required to create a connection and setup the access time to allow Talend read the data. If Talend access out side of the allow time, it will not get the file.

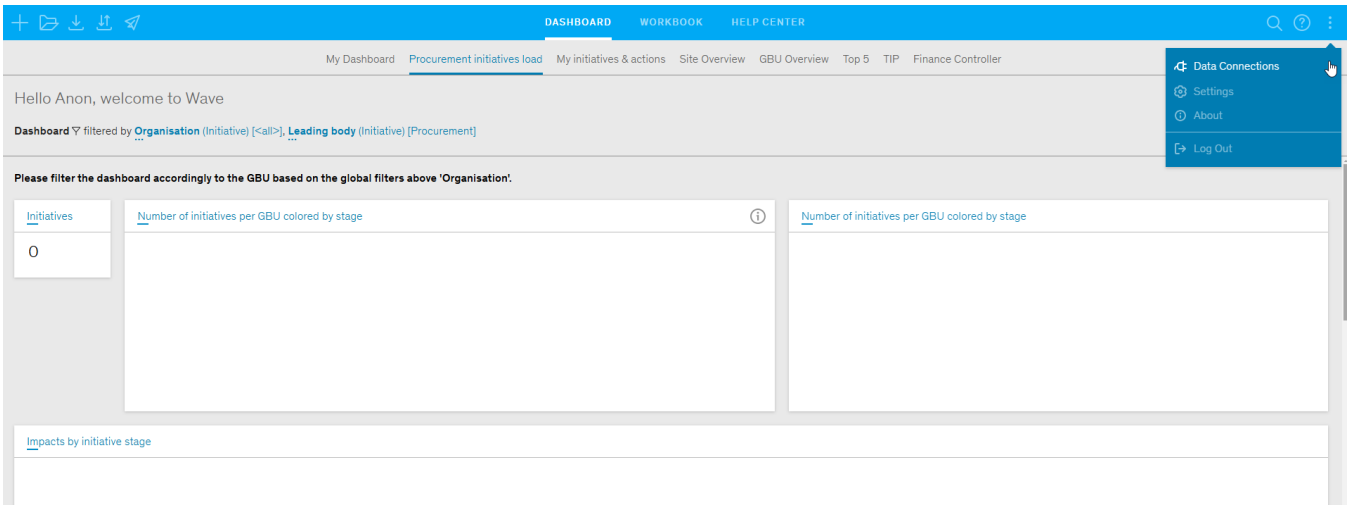
The password is also required to reset every 6 months since the password will be expired

## Wave Connection

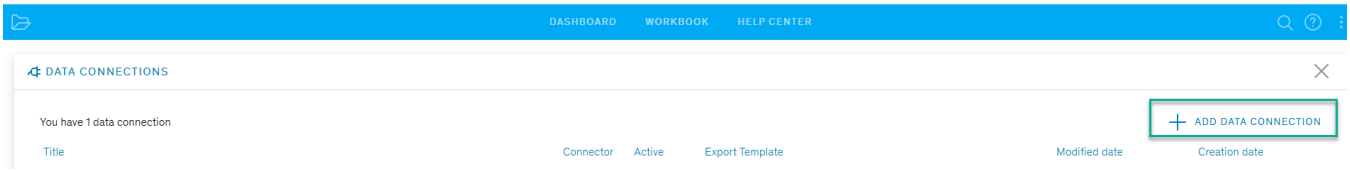
### How to create connection

1. Login https://solvay.mckinseywave.com/login/auth

2. Go to Data Connection



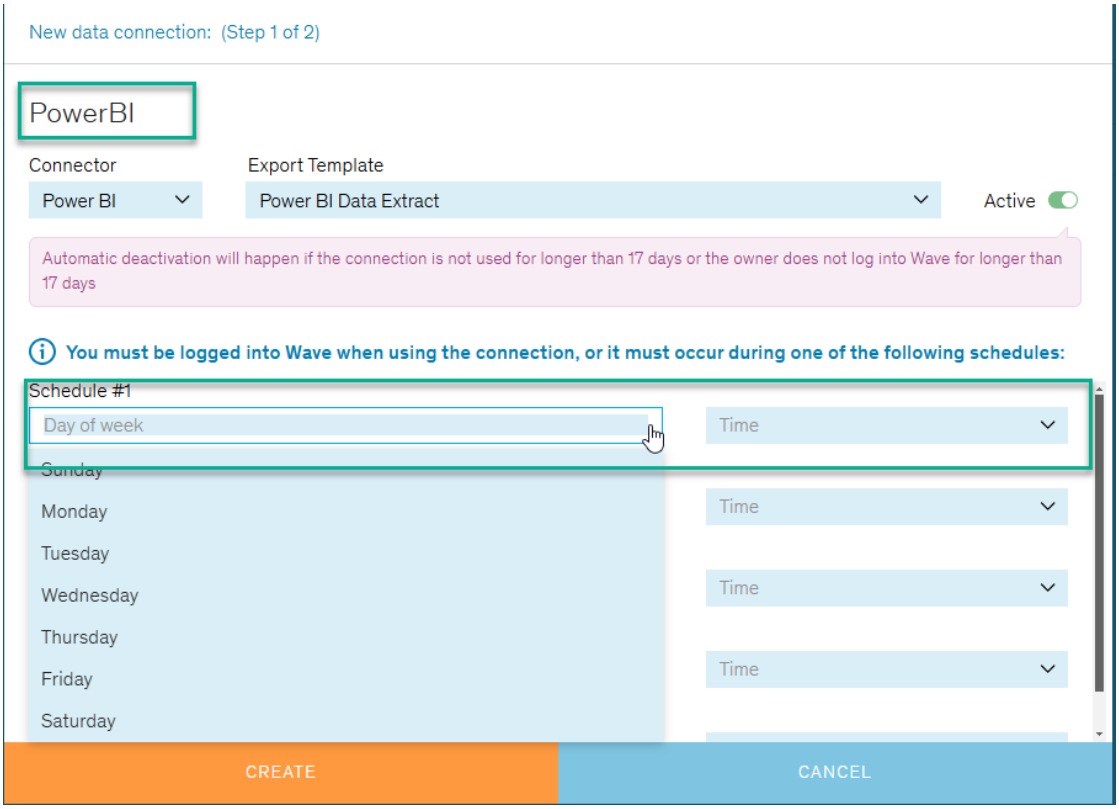

3. Click on ADD DATA CONNECTION




4. Enter the name of the connection and data/time that allow to access.

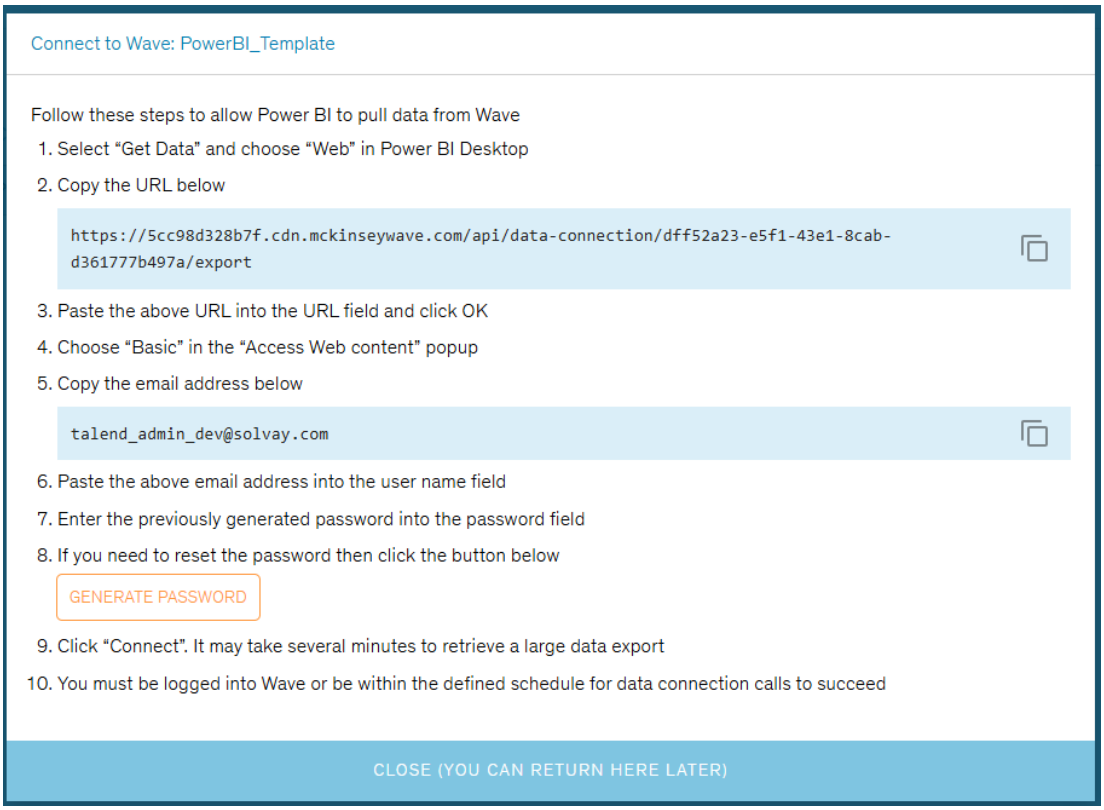
If not set schedule, it is required to login on the web in order to allow the access.

The time is GMT+1 time zone




5. It will generate URL endpoints, username and password. Click on generate password and save it.

Please note that each password will be expired every 6 months and the interface is required to access at least every 17 days, otherwise, the connection will be deactivated.

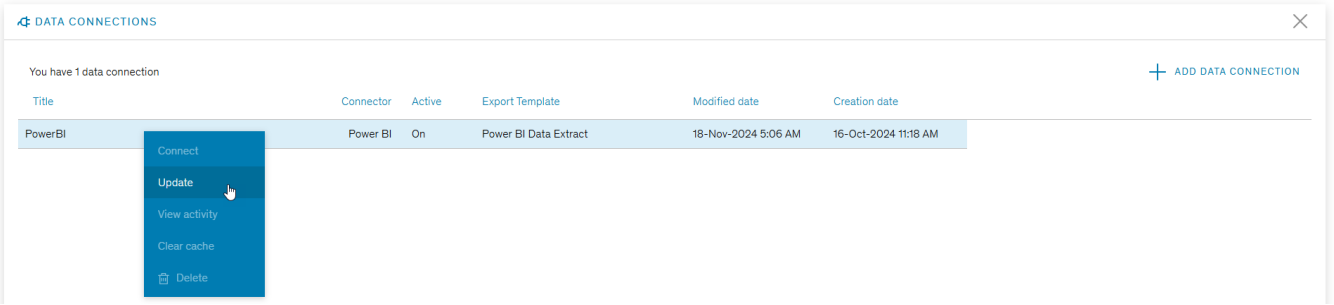



## How to change the access time

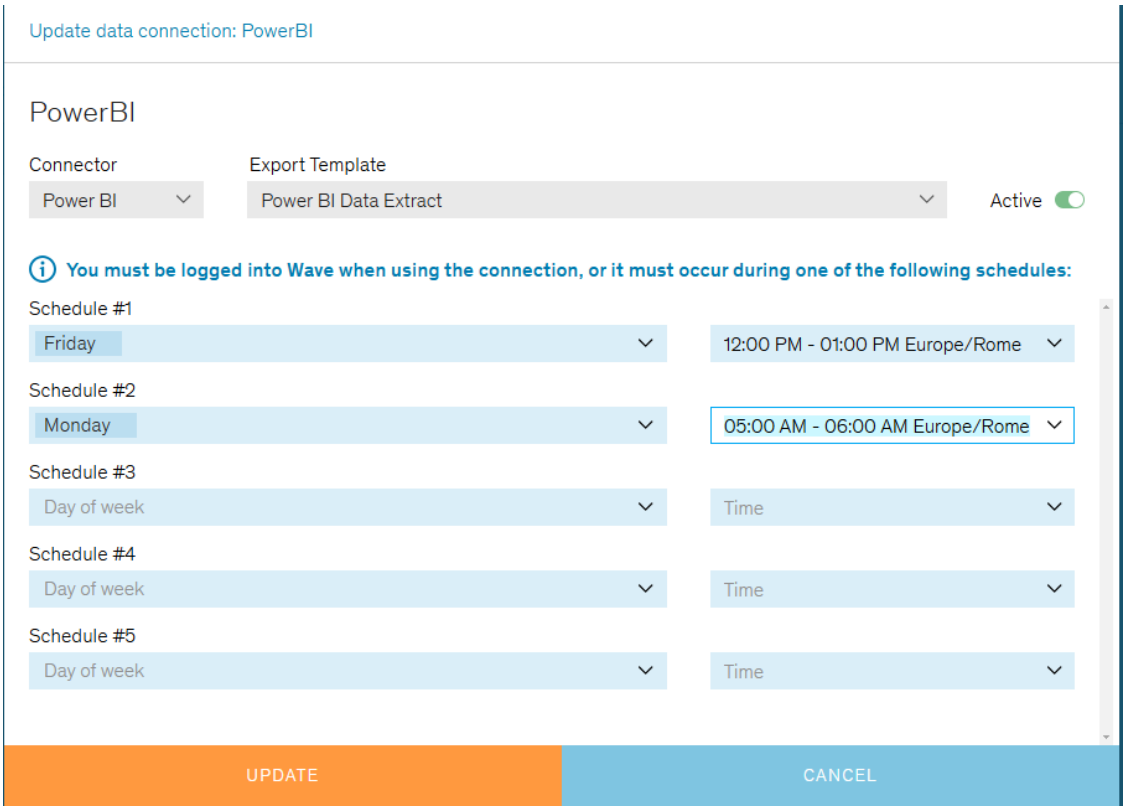
1. Login https://solvay.mckinseywave.com/login/auth

2. Go to Data Connections same as create connection

3. Right click on the connection that you want to change and select Update




4. Change date/time and click UPDATE



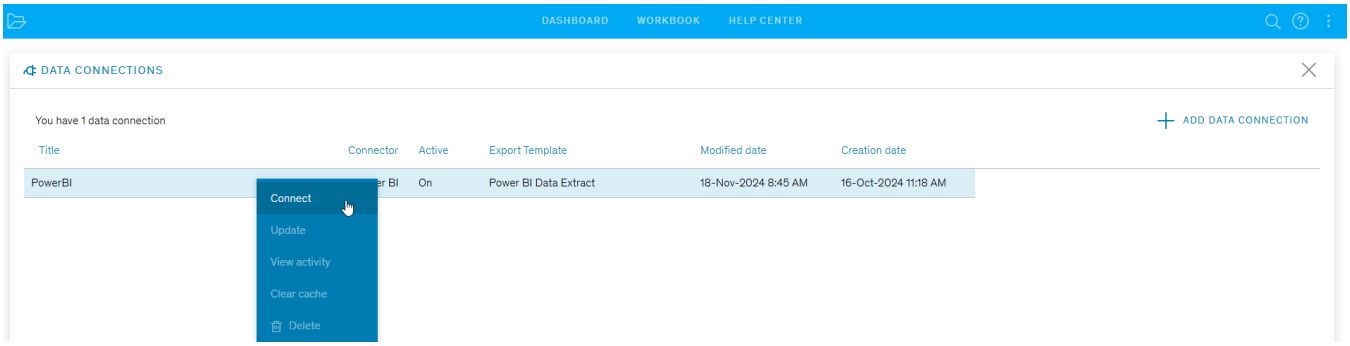

5. In case of the connector is deactivated, just mark it active here and keep login to the Wave in order to avoid deactivate.

## How to reset password

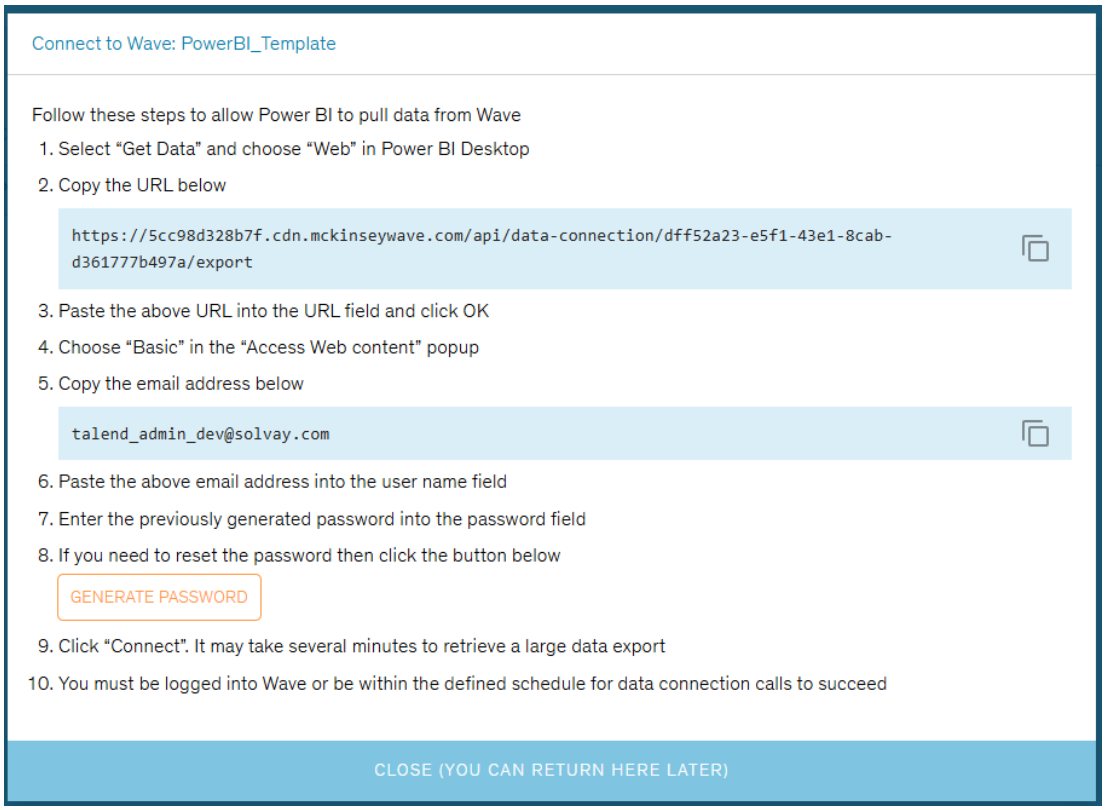
1. Login https://solvay.mckinseywave.com/login/auth

2. Go to Data Connections same as create connection

3. Right click on the connection that you want to change and select Connect




4. Click on reset password



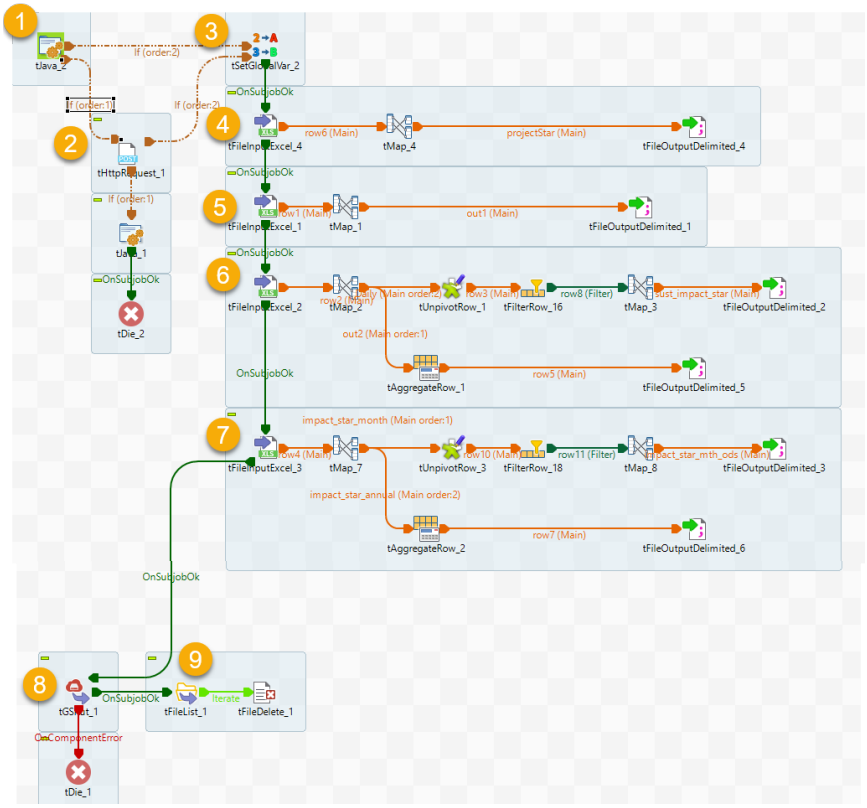

# Talend

The Talend project is on DATA_OCEAN_DOMAIN_INDUSTRIAL

The project will required 4 sheets from the file output from the API. Then, in order to improve the performance of the loading and the size of data, the impactStar and SustainabilityImpactStar sheet are normalize into 2 tables

## Detail job

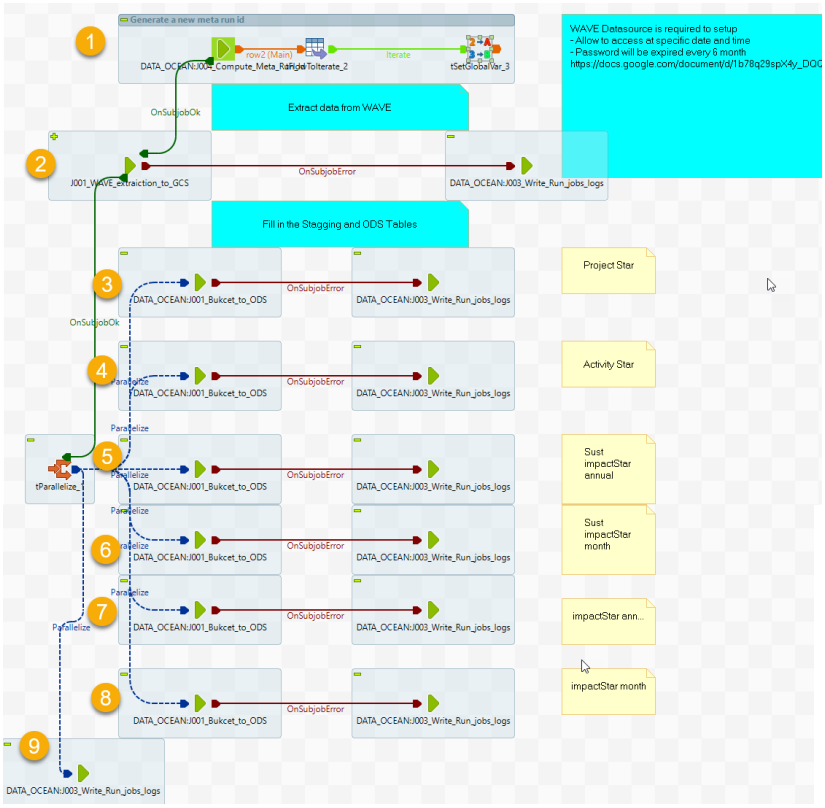
- J001_WAVE_extraiction_to_GCS




1. tJava to check l_VAR_WAVE_reload, true will get the file wave.xlsx, false will get from Wave API
2. tHttpRequest to call Wave api in case l_VAR_WAVE_reload = false
    a. if return code = 200, the api will generate the file wave.xlsx
    b. if return code != 200, the the email will be sent to inform DataOps
3. tSetGlobalVar to set current date for the snapshot value of this loading
4. tFileInputExcel4 to read wave.xlsx on projectStar Sheet, then transform with tMap and save file to "WAV_IND_0000_0000_F001_[yyyyMMddHHmmss]_0000_F_D_project_star.csv"
5. ~~tFileInputExcel1 to read wave.xlsx on activityStar Sheet, then transform with tMap and save file to "WAV_IND_0000_0000_F002_[yyyyMMddHHmmss]_0000_F_D_activity_star.csv"~~
6. tFileInputExcel2 to read wave.xlsx on sustainabilityImpactStar Sheet, then spit into 2 files
    a. month : unpivot by changing column (period) to row and remove the line which does not have data blank or 0 and save to "WAV_IND_0000_0000_F003_[yyyyMMddHHmmss]_0000_F_D_sust_impact_star_month.csv"
    b. annual : keep all columns except the Purpose and period and aggregate the row to reduce duplicate rows and save file to "WAV_IND_0000_0000_F003_[yyyyMMddHHmmss]_0000_F_D_sust_impact_star_annaul.csv"
7. tFileInputExcel3 to read wave.xlsx on impactStar Sheet, then spit into 2 files
    a. month : unpivot by changing column (period) to row and remove the line which does not have data blank or 0 and save to "WAV_IND_0000_0000_F004_[yyyyMMddHHmmss]_0000_F_D_impact_star_month.csv"
    b. annual : keep all columns except the Purpose and period and aggregate the row to reduce duplicate rows and save file to "WAV_IND_0000_0000_F004_[yyyyMMddHHmmss]_0000_F_D_impact_star_annaul.csv"
8. Upload all 6 files to GCS
9. Delete the 6 files from Talend server

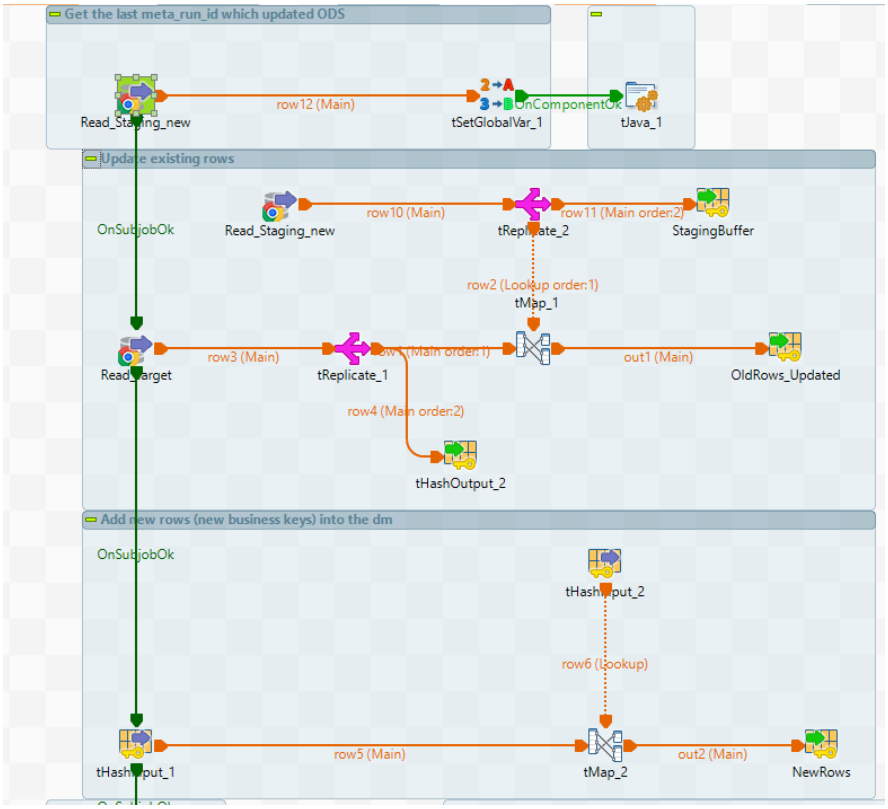
## Flow job

- F001_WAVE_TO_BQ




1. Setup meta_run_id and filename of the output file
2. Call the detail job and pass parameters
3. Call the standard job to upload the projectStar files from GCS to ODS
4. ~~Call the standard job to upload the activityStar files from GCS to ODS~~
5. Call the standard job to upload the sustainabilityImpactStar annual files from GCS to ODS
6. Call the standard job to upload the sustainabilityImpactStar month files from GCS to ODS
7. Call the standard job to upload the impactStar annual files from GCS to ODS
8. Call the standard job to upload the impactStar month files from GCS to ODS
9. Update the log.

- F011_DIM_wave_project_star_SCD1




Select only last loading in ODS update to DIM_wave_project_star

## Access rights

It is required to get access from WAVE admin ednamaria.coan@solvay.com

# Source

- https://solvay.mckinseywave.com/login/auth prj-data-dm-industrial-[env]
- The API URL will be configured on Wave connection and the URL will keep in parameter l_VAR_WAVE_url_api

## Format

- columnar format (Excel - .xlsx)

# Destination

## Data Ocean

- Bucket = cs-ew1-prj-data-dm-industrial-[env]-staging
- DataOean GCP = prj-data-dm-industrial-[env]
- STG Table name = prj-data-dm-industrial-[env].STG.
  - STG_WAV_0000_0000_F001_F_D_project_star
  - ~~STG_WAV_0000_0000_F002_F_D_activity_star~~
  - STG_WAV_0000_0000_F003_F_D_sust_impact_star_annual
  - STG_WAV_0000_0000_F003_F_D_sust_impact_star_month
  - STG_WAV_0000_0000_F004_F_D_impact_star_annual
  - STG_WAV_0000_0000_F004_F_D_impact_star_month
- ODS Table name = prj-data-dm-industrial-[env].ODS
  - ODS_WAV_0000_F001_F_D_project_star
  - ~~ODS_WAV_0000_F002_F_D_activity_star~~
  - ODS_WAV_0000_F003_F_D_sust_impact_star_annual
  - ODS_WAV_0000_F003_F_D_sust_impact_star_month
  - ODS_WAV_0000_F004_F_D_impact_star_annual
  - ODS_WAV_0000_F004_F_D_impact_star_month

- DM Tables name = prj-data-dm-industrial-[env].DM
    - DIM_wave_project_star
    - ~~DIM_wave_activity_star~~

## Product

- Bucket = cs-ew1-prj-data-industrial-dash-[env]-staging
- Product GCP = prj-data-industrial-dash-[dev]
- Dataset = prj-data-industrial-dash-dev.DataOcean  DPL
    - ~~V_ODS_wave_activity_star_fl~~
    - V_ODS_wave_impact_star_annual_fl
    - V_ODS_wave_impact_star_month_fl
    - V_ODS_wave_project_star_fl
    - ~~V_wave_activity_star~~
    - V_wave_impact_star_annual
    - V_wave_impact_star_month
    - V_wave_project_star
    - V_wave_sust_impact_star_annual
    - V_wave_sust_impact_star_month

## Format

- columnar format

## Sizing

ProjectStar around 7000 records

Sust ImpactStart annual around 9000 records

Sust ImpactStart month around 54000 records

ImpactStart annual around 85,000 records

ImpactStart month around 1,226,000 records

## Assessment

Data should be the same as output file and

SELECT distinct snapshot FROM `prj-data-dm-industrial-dev.ODS.ODS_WAV_0000_F004_F_D_impact_star_annual`
order by snapshot desc

The current date of snapshot should be available

# Loading

## 1.1 Incremental Load

### Source to ODS

N/A  Always full load from the file

### ODS to DM

l_VAR_dim_wave_project_star_reload = incremental

## 1.2 Full load

Set l_VAR_WAVE_reload = false

PL_INDUS_WAVE  will load the output excel file from Wave API and save the current time as snapshot field

## 1.3. Reloading data

### Source to ODS

Set l_VAR_WAVE_reload = true  (in case of load historical data, manual file, which is not get from Wave API)

Key users will provide the historical file and load the file manually by rename the file to wave.xlsx on folder \\Acew1dtlndeng02\data\DEV\DATA_OCEAN_DOMAIN_INDUSTRIAL\InOut\WAVE.  The file must have the same sequence of column same as the normal extraction.

### ODS to DM

l_VAR_dim_wave_project_star_reload = QUALIFY ROW_NUMBER() OVER (PARTITION BY snapshot, projectStar ORDER BY meta_ods_insert_date DESC) = 1

## 1.4 Plan to schedule

Weekday at 04:01 AM CET

## 1.5 Timing

The average time expected for  loading: around 20 mins

# Criticality

> *High/Medium/Low*

# Logging

On project prj-data-dm-industrial-[environment]

```
select job.job_name, job.meta_start_date, job.meta_execution_id, logs.meta_run_id, logs.meta_source_system, logs.meta_step, logs.meta_status, logs.meta_num_lines, logs.meta_error_lines from STG.log_tables logs join STG.run_jobs job on logs.meta_run_id = job.meta_run_id
where  logs.meta_run_id in (SELECT meta_run_id FROM STG.run_jobs order by meta_start_date desc limit 100)
and job_name like '%WAVE%'
and meta_start_date >  DATE_SUB(CURRENT_TIMESTAMP(), INTERVAL 24 HOUR)
order by job.meta_start_date desc
```