About Incremental Data Import using TD Workflow

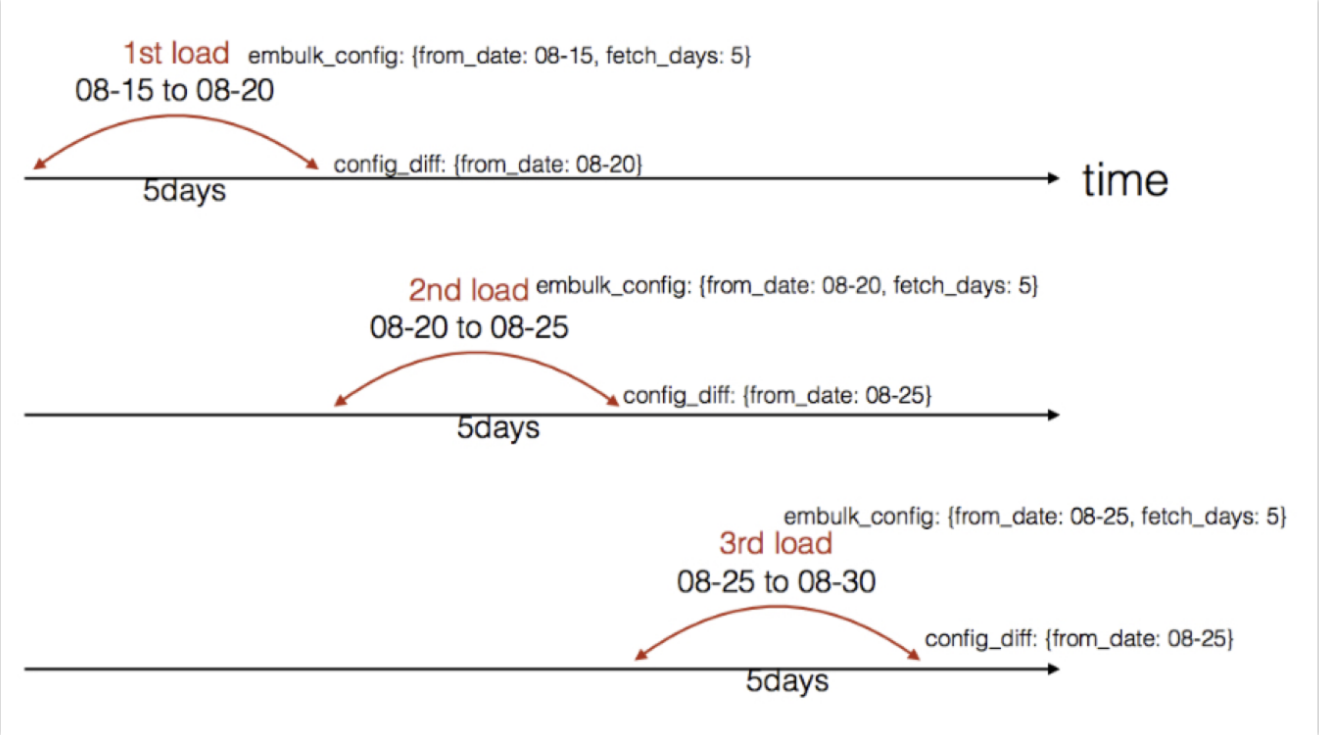
You can specify incremental transfers of data and control when data transfers start as part of managing your data pipeline. Use TD Workflows to ensure query processing steps in Treasure Data run only after the necessary data has been ingested into your account.

In some cases, you'll want to import only the most recently added or modified records. You might want to run incrementally when:

- Tables are too large to regularly re-import the table in its entirety (for example, from large production databases)
- You are running frequent imports of updated data (for example every 15 minutes) to keep the data as fresh as possible
- You want to minimize the number of rows ingested, to make the most efficient use of your Treasure Data account plan's capacity

About Timing, Scheduling, and Incremental Processing

The incremental processing works by keeping track of the column value of the table and records to be imported (for example. a `time` or `ID` column), and then using the highest value imported during the last ingest to start the subsequent ingest. For example:



The `from_date` (can be last_fetched_date or any incremental field) is updated and stored after each execution. New value is used during next run.

For example, using the Mixpanel data connector:

- During the first run, you import all data.
- For subsequent incremental runs, you use the `last_fetched_time` (which is the max ingestion timestamp from the previous run)

About the session_time Variable

Review the concepts of scheduled executions and using the session-time variable: Digdag scheduled execution and session_time.

You create a custom data transfer configuration, using the session_time variable, which is a Digdag workflow variable.

You specify the variable directly in your data connector configuration file.

Use the session_time variable in a workflow to achieve incremental transfer when you need flexibility in the timing, for example, when the system you're pulling data from has late arriving data.

You can use session_time variable as a basis for incremental transfer only when the incremental flow is based on a date.

Determine Your Approach

You can set up incremental input transfers using one of the following:

Approach	Use	Do Not Use
Unique ID based	When you are handling incremental processing logic from IDs or file names, such as using column names. Or, for example, when you want to keep track of the ID of the last record, and then pull from the next record onward at the next ingestion event. When you are not ingesting data on a regular basis. When you want the system to be more self-healing. For example, when you have a single failure event. With self-healing, if an ingest fails, but the next succeeds, the 'diff' calculated will be from the last successful ingestion occurrence.	If the system from which you're pulling batch incremental loads into Treasure Data has "late arriving data." For example, certain SaaS analytic tools make events available through API, but do not guarantee that events become available for querying in the order the events arrive. In this case, the default logic of this approach may miss earlier data that arrives late.
YAML file based	Only when the incremental flow is based on a date. When you need flexibility in the timing. For example, when the system you're pulling data from has late arriving data.	You cannot use this approach if you are trying to run incremental processing based on a non-date column in the data set you're ingesting from.