Using runtime parameters in transfers

When setting up a Cloud Storage or Amazon S3 transfer, the URI and the destination table can both be parameterized, letting you load data from buckets organized by date. These parameters are referred to as *runtime parameters* to distinguish them from query parameters.

When you use runtime parameters in a transfer, you can do the following:

- Specify how you want to partition the destination table

- Retrieve files that match a particular date

Available runtime parameters

When setting up the Cloud Storage or Amazon S3 transfer, you can specify how you want to partition the destination table by using runtime parameters.

Parameter	Template type	Value
run_time	Formatted timestamp	In UTC time, per the schedule. For regularly scheduled transfers, run_time represents the intended time of execution. For example, if the transfer is set to "every 24 hours", the run_time difference between two consecutive queries will be exactly 24 hours—even though the actual execution time might vary slightly. See TransferRun.runTime (/bigquery-transfer/docs/reference/datatransfer/rest/v1/projects.locations.transferConfigs.runs
run_date	Date string	The date of the run_time parameter in the following format: %Y%m%d; for example, "20180101". This format is compatible with ingestion-time partitioned tables.

Templating system

Cloud Storage and Amazon S3 transfers support runtime parameters in the destination table name by using a templating syntax.

Parameter templating syntax

The templating syntax supports basic string templating and time offsetting. Parameters are referenced in the following formats:

- `{run_date}`

- `{run_time[+\-offset]|"time_format"}`

ParameterPurpose

`run_date` This parameter is replaced by the date in format `YYYYMMDD`.

`run_time` This parameter supports the following properties:

> `offset`
> Time offset expressed in hours (h), minutes (m), and seconds (s) in that order.
> Days (d) are not supported.
> Decimals are allowed, for example: `1.5h`.
>
> `time_format`
> A formatting string. The most common formatting parameters are years (%Y), months (%m), and days (%d).
> For partitioned tables, YYYYMMDD is the required suffix - this is equivalent to "%Y%m%d".
>
> Read more about <u>formatting datetime elements</u>
> (https://cloud.google.com/bigquery/docs/reference/standard-sql/functions-and-operators#supported-format-elements-for-datetime)
> .

Usage notes:

- No whitespace is allowed between run_time, offset, and time format.

- To include literal curly braces in the string, you can escape them as '`\{`' and '`\}`'.

- To include literal quotes or a vertical bar in the time_format, such as "`YYYY|MM|DD`", you can escape them in the format string as: '`\"`' or '`\|`'.

Parameter templating examples

These examples demonstrate specifying destination table names with different time formats, and offsetting the run time.

run_time (UTC)	Templated parameter	Output destination table name
2018-02-15 00:00:00	mytable	mytable
2018-02-15 00:00:00	mytable_{*run_time*|*"%Y%m%d"*}	mytable_20180215
2018-02-15 00:00:00	mytable_{*run_time+25h*|*"%Y%m%d"*}	mytable_20180216
2018-02-15 00:00:00	mytable_{*run_time-1h*|*"%Y%m%d"*}	mytable_20180214
2018-02-15 00:00:00	mytable_{*run_time+1.5h*|*"%Y%m%d;%H"*} or mytable_{*run_time+90m*|*"%Y%m%d;%H"*}	mytable_2018021501
2018-02-15 00:00:00	{*run_time+97s*|*"%Y%m%d"*}_mytable_{*run_time*|*"%H%M%s"*}20180215_mytable_000137	

Partitioning options

Cloud Storage and Amazon S3 transfers can write to partitioned or non-partitioned destination tables. There are two types of table partitioning in BigQuery:

- **Tables partitioned by ingestion time**: Tables that are partitioned based on the ingestion time. For Cloud Storage and Amazon S3 transfers, the ingestion time is the transfer's run time.

- **Partitioned tables**: Tables that are partitioned based on a column. The column type must be a TIMESTAMP (/bigquery/docs/reference/standard-sql/data-types#timestamp-type) or DATE (/bigquery/docs/reference/standard-sql/data-types#date-type) column.

If the destination table is partitioned on a column, you identify the partitioning column when you create the destination table and specify its schema. Learn more about creating column-based partitioned tables in Creating and using partitioned tables (/bigquery/docs/creating-column-partitions).

Minutes cannot be specified when partitioning a table.

Partitioning examples

- Table with no partitioning

    - Destination table: `mytable`

- Ingestion-time partitioned table
  (/bigquery/docs/partitioned-tables#ingestion-time_partitioned_tables)

    - Destination table: `mytable$YYYYMMDD`

    - Note that minutes cannot be specified.

- Column-partitioned table (/bigquery/docs/partitioned-tables#date_timestamp_partitioned_tables)

    - Destination table: `mytable`

    - Specify the partitioning column as a `TIMESTAMP` or `DATE` column when you create the table's schema.

Notes on parameter usage

- If you partition your data based on your local timezone, you will need to manually calculate the hour offset from UTC by using the offsetting mechanism in the templating syntax (#templating_system).

- Minutes cannot be specified in parameters.

- Using wildcards for the Cloud Storage URI in combination with parameters on the destination table name is allowed. For more information about wildcards, see Wildcard support for Cloud Storage URIs
  (/bigquery-transfer/docs/cloud-storage-transfer-overview#wildcard-support).

Runtime parameter examples

The following examples show ways to combine the wildcard character and parameters for common use cases. Assume the table's name is `mytable` and the `run_time` is `2018-02-15 00:00:00` (UTC) for all examples.

Transferring Cloud Storage or Amazon S3 data to a non-partitioned table

This use case applies to loading new files from a Cloud Storage or Amazon S3 bucket into a non-partitioned table. This example uses a wildcard in the URI and uses an ad hoc refresh (/bigquery-transfer/docs/cloud-storage-transfer#setting_up_a_refresh_run_of_a_transfer) transfer to pick up new files that are at least one hour old.

Data source	Source URI	Destination table name
Cloud Storage	gs://bucket/*.csv	mytable
Amazon S3	s3://bucket/*.csv	mytable

Loading a snapshot of all data into an ingestion-time partitioned table

In this case, all data in the specified URI is transferred to a table partitioned by today's date. In a refresh (/bigquery-transfer/docs/cloud-storage-transfer#setting_up_a_refresh_run_of_a_transfer) transfer, this configuration picks up files added since the last load and adds them to a particular partition.

Data source	Source URI	Parameterized destination table name	Evaluated destination table name
Cloud Storage	gs://bucket/*.csv	mytable${*run_time*|*"%Y%m%d"*}	mytable$20180215
Amazon S3	s3://bucket/*.csv	mytable${*run_time*|*"%Y%m%d"*}	mytable$20180215

This use case transfers today's data into a table partitioned on today's date. This example also applies to a refresh (/bigquery-transfer/docs/cloud-storage-transfer#setting_up_a_refresh_run_of_a_transfer) transfer that retrieves newly added files that match a certain date and loads the data into the corresponding partition.

Data source	Parameterized URI	Parameterized destination table name	Evaluated URI	Evaluated destination table name
Cloud Storage	gs://bucket/events-{*run_time*|*"%Y%m%d"*}/*.csv	mytable${*run_time*|*"%Y%m%d"*}	gs://bucket/events-20180215/*.csv	mytable$20180215

Data source	Parameterized URI	Parameterized destination table name	Evaluated URI	Evaluated destination table name
Amazon S3	s3://bucket/events-{*run_time*|*"%Y%m%d"*}/*.csv	mytable${*run_time*|*"%Y%m%d"*}	s3://bucket/events-20180215/*.csv	mytable$20180215

What's next

- Learn about [setting up a Cloud Storage transfer](/bigquery-transfer/docs/cloud-storage-transfer)
  (/bigquery-transfer/docs/cloud-storage-transfer)

- Learn about [setting up an Amazon S3 transfer](/bigquery-transfer/docs/s3-transfer) (/bigquery-transfer/docs/s3-transfer)

- Learn more about the [BigQuery Data Transfer Service](/bigquery-transfer/docs/transfer-service-overview)
  (/bigquery-transfer/docs/transfer-service-overview)