

File Settings

When you generate file-based results, you can configure the filename, storage format, compression, number of files, and the updating actions in the right-hand panel.

Close (X)

Create a new file [Parameterize destination](#)

POS-r01

Output Directory
/1

Data Storage Format
CSV

Create new file every run
Create a new file with an incremental number appended to the name (e.g. POS-r01_2.csv)

Append to this file every run
Create it if it doesn't exist.

Replace this file every run
Create it if it doesn't exist.

[Less options](#) ^

Output options

Include headers as first row on creation

Include quotes

Cancel Add

Figure: Output File Settings

Configure the following settings.

1. **Create a new file:** Enter the filename to create. A filename extension is automatically added for you, so you should omit the extension from the filename.
2. **Output directory:** Read-only value for the current directory. To change it, navigate to the proper directory.

NOTE: During job execution, a canary file is written for each set of results to validate the path. For datasets with parameters, if the path includes folder-level parameterization, a separate folder is created for each parameterized path. During cleanup, only the the canary files and the original folder path are removed. The parameterized folders are not removed. This is a known issue.

3. **Data Storage Format:** Select the output format you want to generate for the job.
 - a. **Avro:** This open source format is used widely for data serialization and data exchange between systems. This format is used to support data serialization within a Hadoop environment.
 - b. **CSV and JSON:** These formats are supported for all types of imported datasets and all running environments.

NOTE: JSON-formatted files that are generated by Trifacta are rendered in JSON Lines format, which is a single line per-record variant of JSON. For more information, see <http://jsonlines.org>.

- c. **Parquet:** This format is a columnar storage format.
- d.

HYPHER: Choose HYPHER to generate results that can be imported into Tableau.

If you have created a Tableau Server connection, you can write results to Tableau Server or publish them after they have been generated in Hyper format.

NOTE: If you encounter errors generating results in Hyper format, additional configuration may be required. See *Supported File Formats*.

- e. For more information, see *Supported File Formats*.
4. **Publishing action:** Select one of the following:

NOTE: If multiple jobs are attempting to publish to the same filename, a numeric suffix (N) is added to the end of subsequent filenames (e.g. `filename_1.csv`).

NOTE: If a single user executes two jobs with the same output settings except for different methods (e.g. create vs. replace) on the same output destination, the generated results and potential error conditions are unpredictable. Please wait for the first job to complete execution before changing the configuration for the second job.

- a. **Create new file every run:** For each job run with the selected publishing destination, a new file is created with the same base name with the job number appended to it (e.g. `myOutput_2.csv`, `myOutput_3.csv`, and so on).
- b. **Append to this file every run:** For each job run with the selected publishing destination, the same file is appended, which means that the file grows until it is purged or trimmed.

NOTE: When publishing single files to S3 or WASB, the `append` action is not supported.

NOTE: When appending data into a Hive table, the columns displayed in the Transformer page must match the order and data type of the columns in the Hive table.

NOTE: Compression of published files is not supported for an `append` action.

- c. **Replace this file every run:** For each job run with the selected publishing destination, the existing file is overwritten by the contents of the new results.

5. **More Options:**

- a. **Include headers as first row on creation:** For CSV outputs, you can choose to include the column headers as the first row in the output. For other formats, these headers are included automatically.

NOTE: Headers cannot be applied to compressed outputs.

- b. **Include quotes:** For CSV outputs, you can choose to include double quote marks around all values, including headers.
- c. **Delimiter:** For CSV outputs, you can enter the delimiter that is used to separate fields in the output. The default value is the global delimiter, which you can override on a per-job basis in this field.

Tip: If needed for your job, you can enter Unicode characters in the following format: `\uXXXX`.

NOTE: The Spark running environment does not support use of multi-character delimiters for CSV outputs. You can switch your job to a different running environment or use single-character delimiters. For more information on this issue, see <https://issues.apache.org/jira/browse/SPARK-24540>.

- d. **Single File:** Output is written to a single file. Default setting for smaller, file-based jobs or for jobs executed on the Trifacta Server.
 - e. **Multiple Files:** Output is written to multiple files. Default setting for larger file-based jobs or for jobs executed on in a remote, cluster-based running environment.
6. **Compression:** For text-based outputs, compression can be applied to significantly reduce the size of the output. Select a preferred compression format for each format you want to compress.

NOTE: If you encounter errors generating results using Snappy, additional configuration may be required. See *Supported File Formats*.

7. To save the publishing action, click **Add**.