

# News

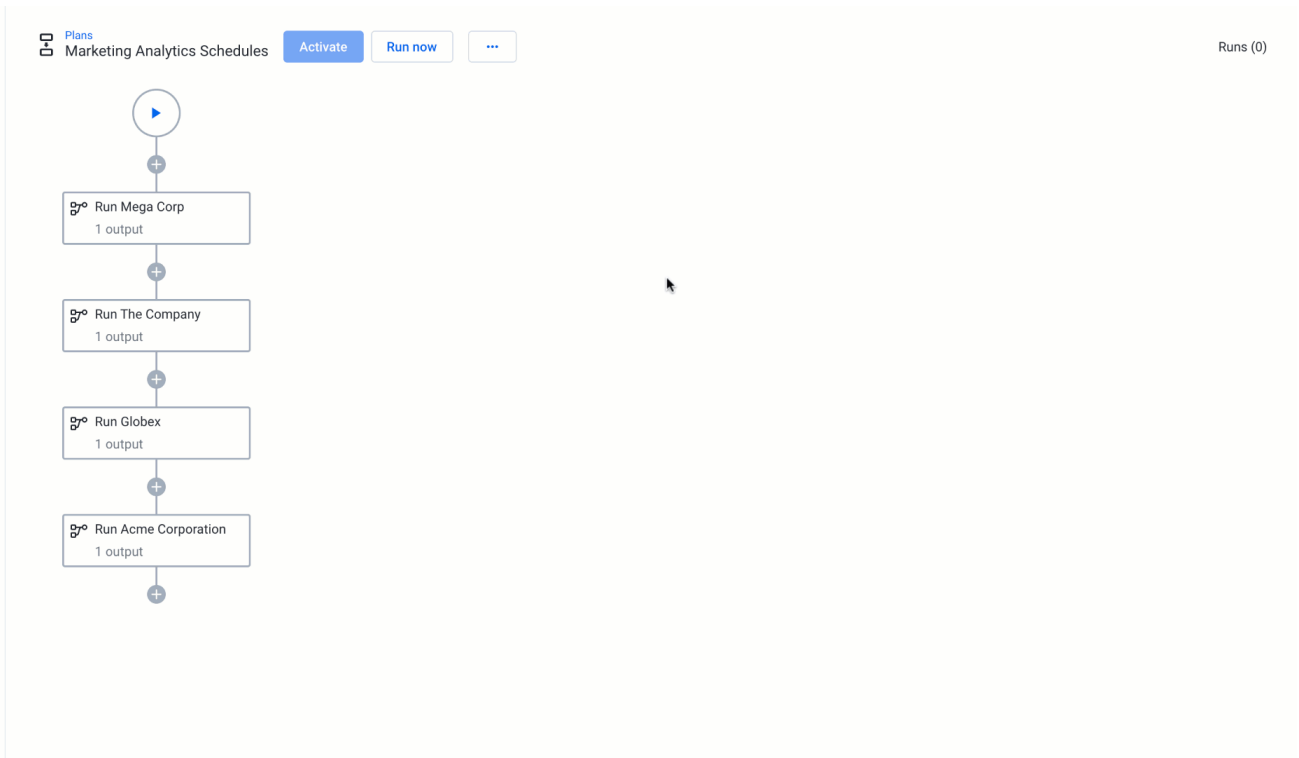
## Release 7.1

May 4, 2020

Trifacta® started out as a company focused on creating a new and easy-to-use experience for structuring, cleaning, blending, and aggregating data. Over the years, we saw more and more of our customers using Trifacta not only to clean data, but to deploy data preparation recipes into data pipelines that feed analytics processes, AI workloads and more. The feedback we got is that legacy tools for data pipelining were rigid and hard to use. Trifacta was not only useful for self-service data preparation, but the intelligent, visual interface made it easier for users to discover and clean data quality issues, uncover and extract new insights in data, and build and deploy data pipelines faster. A big focus of our development work over the last year has been to expand our capabilities in automation and data pipelining, and this release has several features that enhance the ability to create robust and agile data pipelines.

### Pipeline Orchestration with Plans

In the Spring 2020 release, Trifacta has a new core feature – Plans. Plans provide a command center within the UI to orchestrate data pipelines. Plans allow for the sequencing of flows and jobs, bulk scheduling and running of jobs and organization of job runs, webhooks, and other tasks.



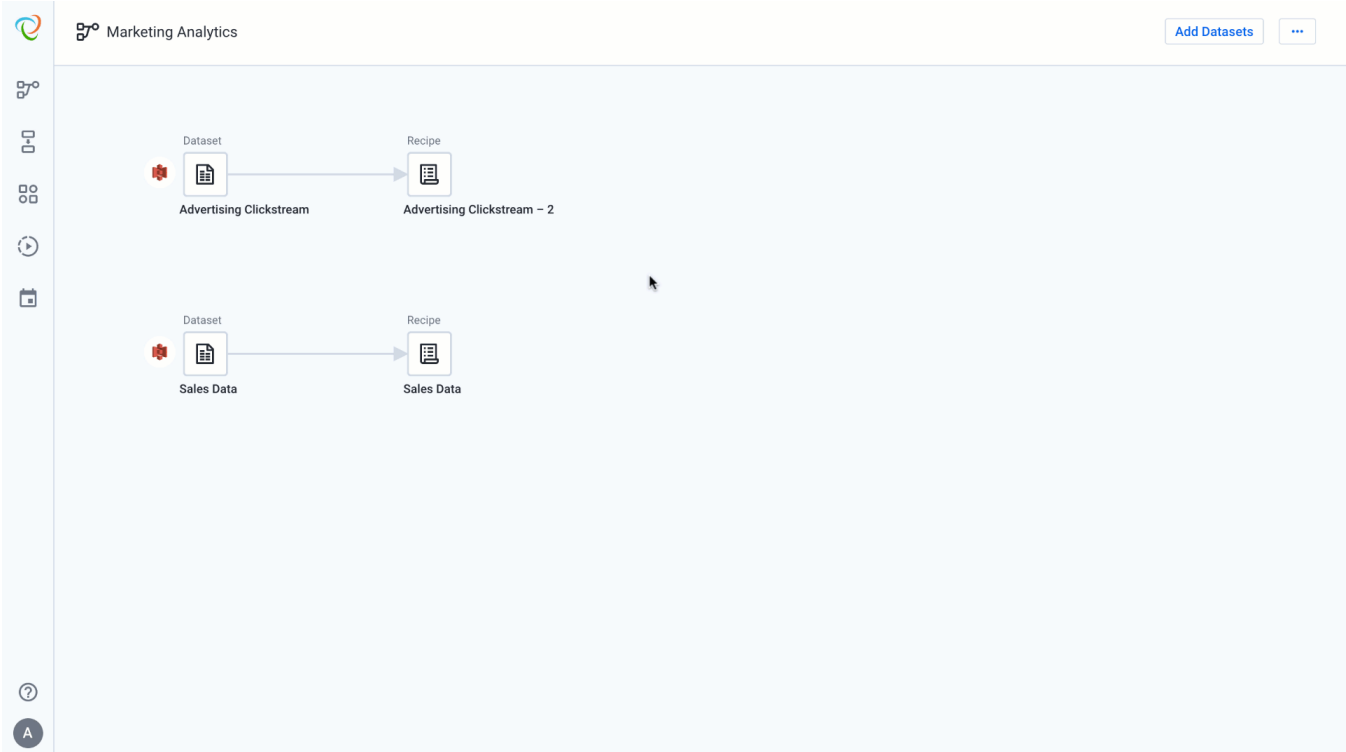
**Figure: Plan View**

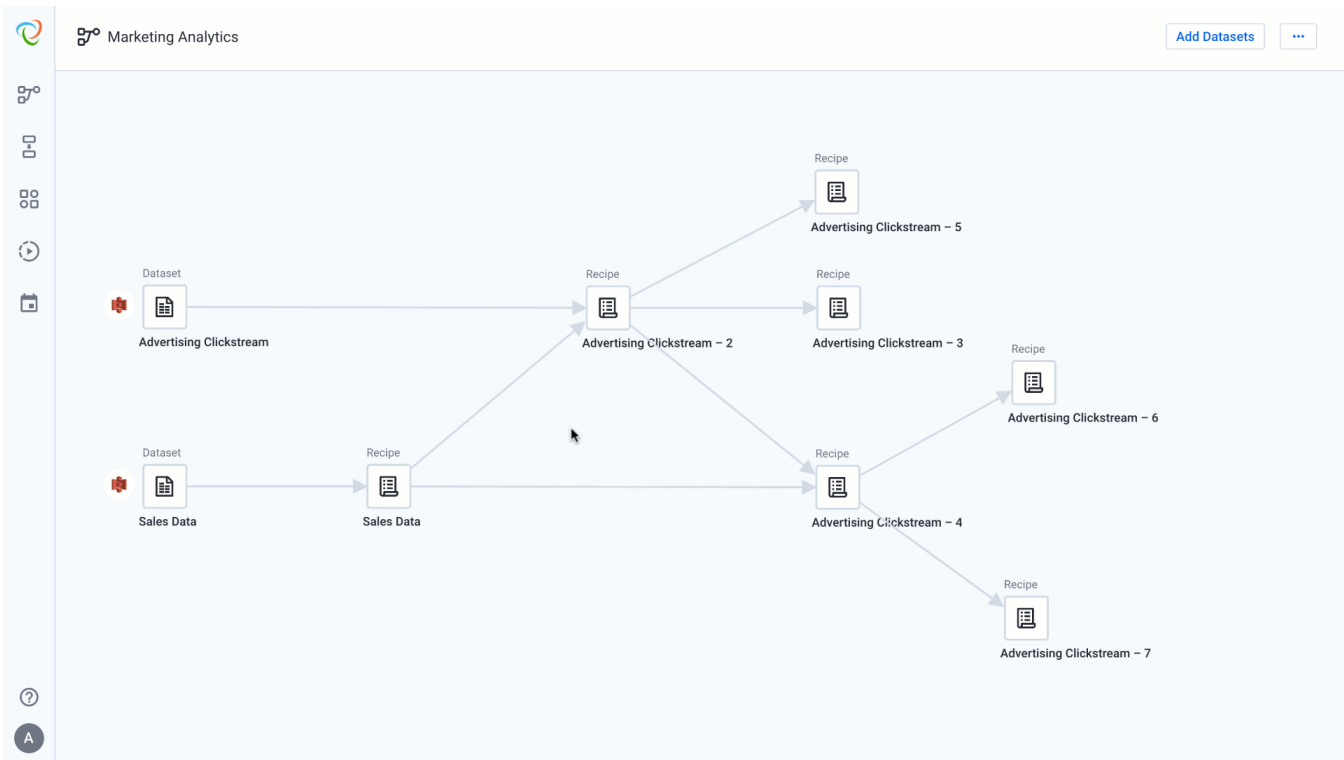
We are really excited to see where we can go with this feature and will be continuing to improve it over time to make it more flexible and incorporate more areas of the product in Plans.

## Enhanced Flow View

**NOTE:** This feature is in Beta release.

Available as a Beta release on Trifacta Wrangler Enterprise only, enhanced flow view improves the user experience when working within a flow. Users can now drag and drop datasets, recipes, and outputs, can zoom in and zoom out, and leave comments and notes within the flow. You can also utilize multi-select with command /ctrl + drag. Additionally, users can now initiate joins and unions within the flow view. To check it out, you can toggle between the new flow view and the old flow view from the Flow View context menu.





**Figure: Enhanced flow view**

Enhanced flow view will be available across all product editions in the next release.

## Flow Parameters

Also new in the Spring 2020 release, flow parameters expand the possibility of templated recipes and flows in Trifacta. Users can create dynamic filters, if/then statements, patterns, flags, etc. that allow them to control the output of a flow from one job to the next, without having to modify steps in the recipe each time. Parameters can be added to steps like filters, new formulas, extracts, and replaces.

The screenshot shows a data table with the following columns: RBC, event\_type, RBC, event\_subtype, event\_time, #, advertiser\_id, #, and c. The table contains rows with event types like 'view' and 'click' and advertiser IDs like '164332'. A 'Filter rows' dialog is open on the right, with the following settings:

- Condition: Contains
- Column: RBC event\_type
- Pattern to match: \$Event Type
- Action: Keep matching rows

**Figure: Insert flow parameters into your recipe steps**

After a parameter is added, users then have control at time of job execution to choose the value for their output. For example, in the screenshot we can toggle between “click” and “view”, which will either keep rows where event\_type equals “click” or “view”:

The 'Run Job' dialog box shows the following sections:

- Running Environment:** Photon (Trifacta Server) and Spark.
- Options:** Profile results is checked.
- Publishing Actions:** Create-CSV action with location hdfs://trifacta/queryResults/admin@trifacta.local/AdvertisingClickstream\_%28%29\_\_2.csv and settings no compression, single file, with headers, with quotes, with delimiter.
- Variables:** A table with columns Name, Value, and Source. The 'Event Type' variable is set to 'click' and has a source of 'this flow'.

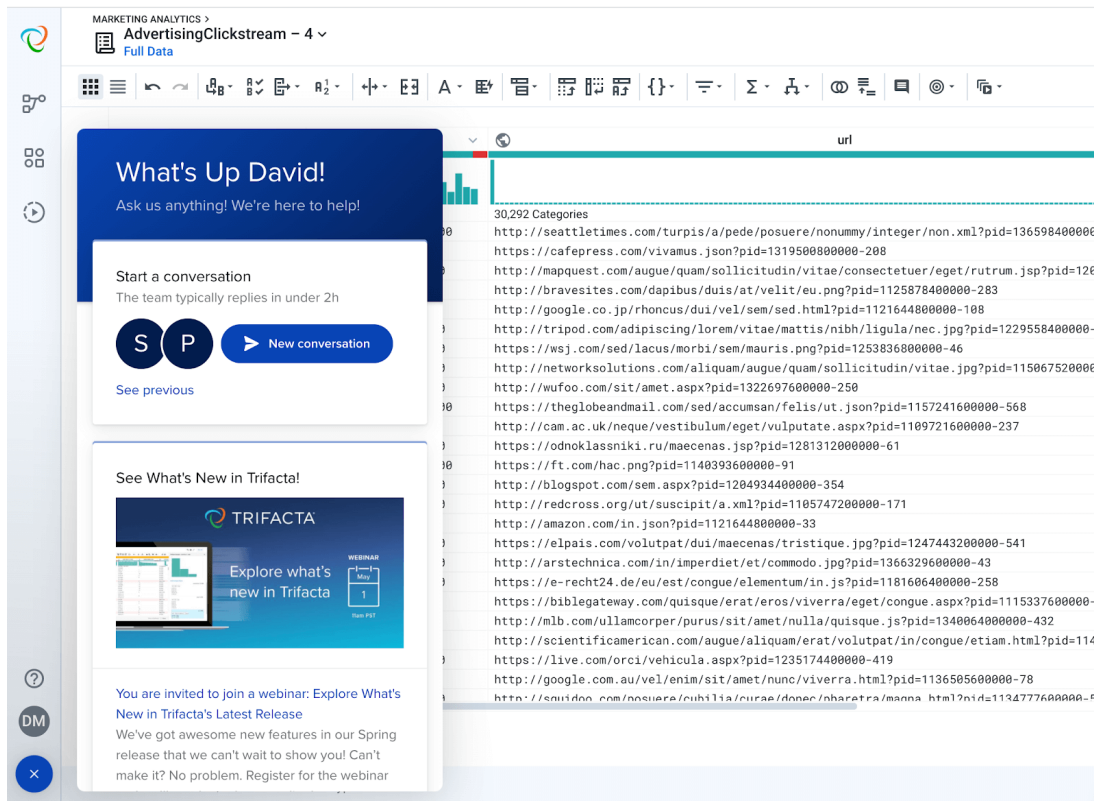
**Figure: Apply flow parameter overrides at job run time**

Parameters can be passed in manually or by API calls. Flow parameters improve reuse within the product, because recipes can now take in different values depending on the underlying data the recipe is being used for, adding to the vision of wrangle once, use everywhere. Flow parameters should reduce the need for copies of flows and recipes, and they also allow users to pass in data from third party systems via the API, adding additional flexibility to the product.

### Talk to Trifacta Experts Live

If you have been using Trifacta recently, you may have noticed a new chat widget in the bottom left of the UI. Having difficulty figuring out what step you want to use? Running into a problem or just in need of additional support? Start a conversation with us to get help immediately!

**NOTE:** For more information on enabling this feature in your Trifacta deployment, please contact *Trifacta Support*.



**Figure: Chat with a product expert**

## Tableau Server Publishing Enhancements – Support for Hyper

The Spring 2020 release brings improvements to direct publishing to Tableau Server. Trifacta now supports publishing in Hyper format to Tableau Server, which should provide faster performance, easier configuration, and improved automation through scheduling.

**NOTE:** Support for publishing to TDE had already been deprecated by Tableau. It is likely to be deprecated in Trifacta in a future release. Please switch to using Hyper format.

## New Transformation Functions

The Spring 2020 release brings a new set of Date functions, covering date aggregation functions, timezone conversion functions, workday functions with regionality, and more. The functions are all listed in the table below:

The screenshot shows the Tableau interface with a data table and a 'New formula' dialog box. The dialog box contains the following information:

- Formula type:** Single row formula
- Formula:** `CONVERTTOUTC(TimeStamp, "America/New_York")`
- New column name:** Insert new name

The background table has the following columns: #, FMID, TimeStamp, converttoutc\_TimeStamp, RBC, Item, ##, and Price. The 'converttoutc\_TimeStamp' column is highlighted in yellow.

**Figure: New Wrangle functions**

### Date calculation functions:

Function Name	Description
<i>MINDATE Function</i>	Computes the minimum value found in all row values in a Datetime column.
<i>MAXDATE Function</i>	Computes the maximum value found in all row values in a Datetime column.
<i>MODEDATE Function</i>	Computes the most frequent (mode) value found in all row values in a Datetime column.

### Work day functions:

Function Name	Description
<i>NETWORKDAYS Function</i>	Calculates the number of working days between two specified dates, assuming Monday - Friday workweek. Optional list of holidays can be specified.
<i>NETWORKDAYS INTL Function</i>	Calculates the number of working days between two specified dates. Optionally, you can specify which days of the week are working days as an input parameter. Optional list of holidays can be specified.
<i>WORKDAY Function</i>	Calculates the work date that is before or after a start date, as specified by a number of days. A set of holiday dates can be optionally specified.
<i>WORKDAYINTL Function</i>	Calculates the work date that is before or after a start date, as specified by a number of days. You can also specify which days of the week are working days and a list of holidays via parameters.

### Time zone conversion functions:

Function Name	Description
<i>CONVERTFROM UTC Function</i>	Converts Datetime value to corresponding value of the specified time zone. Input can be a column of Datetime values, a literal Datetime value, or a function returning Datetime values.
<i>CONVERTTOUT C Function</i>	Converts Datetime value in specified time zone to corresponding value in UTC time zone. Input can be a column of Datetime values, a literal Datetime value, or a function returning Datetime values.
<i>CONVERTTIME ZONE Function</i>	Converts Datetime value in specified time zone to corresponding value second specified time zone. Input can be a column of Datetime values, a literal Datetime value, or a function returning Datetime values.

These functions make it much easier to work with dates in a single step, rather than relying on complex transformations to do timezone conversion or even user defined functions. These all now come out of the box!

For the full Release 7.1 release notes, see *Release Notes 7.1*.

For more information on previous releases, see *Past Releases*.