Trifacta Release Notes

Version: 6.0.2
Doc Build Date: 07/11/2019

For third-party license information, please select **About Trifacta** from the User menu.

Release Notes

This section contains release notes for published versions of Trifacta® Wrangler Enterprise.

Topics:

- *Changes to System Behavior*
  - *Changes to the Language*
  - *Changes to the APIs*
  - *Changes to Configuration*
  - *Changes to the Command Line Interface*
  - *Changes to the Admin Settings Page*
  - *Changes to the Object Model*
  - *Improvements to the Type System*
  - *Changes to the User-Defined Functions*
- *Release Notes 6.0*
- *Release Notes 5.1*
- *Release Notes 5.0*
- *Release Notes 4.2*
- *Release Notes 4.1*
- *Release Notes 4.0*
- *Release Notes 3.2*

Changes to System Behavior

The following pages contain information about changes to system features, capabilities, and behaviors in this release.

Topics:

- *Changes to the Language*
- *Changes to the APIs*
- *Changes to Configuration*
- *Changes to the Command Line Interface*
- *Changes to the Admin Settings Page*
- *Changes to the Object Model*
- *Improvements to the Type System*
- *Changes to the User-Defined Functions*

Changes to the Language

Contents:

- *Release 6.0*
  - *New Functions*
  - *Changes to LIST* inputs*
  - *Renamed functions*
  - *FILL Function has new before and after parameters*
- *Release 5.9*
  - *New functions*
- *Release 5.8*
  - *File lineage information using source metadata references*
  - *New math and statistical functions for arrays*
- *Release 5.7*
  - *WEEKNUM function now behaves consistently across running environments*
- *Release 5.6*

- URLPARAMS function returns null values
- *Release 5.1*
  - *Wrangle now supports nested expressions*
  - *SOURCEROWNUMBER function generates null values consistently*
  - *New Functions*
- *Release 5.0.1*
  - *RAND function generates true random numbers*
- *Release 5.0*
  - *Required type parameter*
  - *Deprecated aggregate transform*
  - *New search terms*
  - *Support for <> operator*
  - *ROUND function takes optional number of digits*
  - *New Functions*
- *Release 4.2.1*
- *Release 4.2*
  - *New Filter transform*
  - *New Case transform*
  - *Rename transform now supports multi-column rename*
  - *Drop specified columns or drop the others*
  - *New string comparison functions*
  - *NOW function returns 24-hour time values*
  - *New Transforms*
  - *New Functions*
- *Release 4.1*
  - *Standardization page and transform have been removed*
  - *Syntax Changes*
- *Release 4.0.1*
  - *Map data type is now Object data type*
  - *Splitrows transform now permits specifying of quote escaping character*
- *Release 4.0*
  - *Script steps displayed in natural language*
  - *set, derive, and window transforms can now perform any type of computation*
  - *Multi-column input support*
  - *Syntax Changes*
  - *Execution Changes*
- *Release 3.2.1*
  - *Syntax Changes*
  - *Execution Changes*
- *Release 3.2*
  - *Syntax Changes*
  - *Deprecated Items*

The following changes have been applied to Wrangle in this release of Trifacta® Wrangler Enterprise.

Release 6.0

New Functions

Function Name	Description
ARRAYINDEXOF Function	Computes the index at which a specified element is first found within an array. Indexing is left to right.
ARRAYRIGHTINDEXOF Function	Computes the index at which a specified element is first found within an array, when searching right to left. Returned value is based on left-to-right indexing.

ARRAYSLICE Function	Returns an array containing a slice of the input array, as determined by starting and ending index parameters.
ARRAYMERGEELEMENTS Function	Merges the elements of an array in left to right order into a string. Values are optionally delimited by a provided delimiter.

Changes to LIST* inputs

The following LIST-based functions have been changed to narrow the accepted input data types. In previous releases, any data type was accepted for input, which was not valid for most data types.

In Release 6.0 and later, these functions accept only Array inputs. Inputs can be Array literals, a column of Arrays, or a function returning Arrays.

> **NOTE:** You should references to these functions in your recipes.

LIST* Functions
LISTAVERAGE Function
LISTMIN Function
LISTMAX Function
LISTMODE Function
LISTSTDEV Function
LISTSUM Function
LISTVAR Function

Renamed functions

The following functions have been renamed in Release 6.0.

Release 5.9 and earlier	Release 6.0 and later
LISTUNIQUE Function	*UNIQUE Function*

FILL Function has new before and after parameters

Prior to Release 6.0, the FILL function replaced empty cells with the most recent non-empty value.

In Release 6.0, `before` and `after` function parameters have been added. These parameters define the window of rows before and after the row being tested to search for non-empty values. Within this window, the most recent non-empty value is used.

The default values for these parameters are `-1` and `0` respectively, which performs a search of an unlimited number of preceding rows for a non-empty value.

> **NOTE:** Upon upgrade, the FILL function retains its preceding behavior, as the default values for the new parameters perform the same unlimited row search for non-empty values.

For more information, see *FILL Function*.

Release 5.9

New functions

The following functions can now be applied directly to arrays to derive meaningful statistics about them.

Function	Description
ARRAYSORT Function	Sorts array values in the specified column, array literal, or function that returns an array in ascending or descending order.
TRANSLITERATE Function	Transliterates Asian script characters from one script form to another. The string can be specified as a column reference or a string literal.

Release 5.8

File lineage information using source metadata references

Beginning in Release 5.8, you can insert the following references into the formulas of your transformations. These **source metadata references** enable you to continue to track file lineage information from within your datasets as part of your wrangling project.

> **NOTE:** These references apply only to file-based sources. Some additional limitations may apply.

reference	Description
`$filepath`	Returns the full path and filename of the source of the dataset.
`$sourcerownumber`	Returns the row number for the current row from the original source of the dataset. > **NOTE:** This reference is equivalent to the SOURCEROWNUMBER function, which is likely to be deprecated in a future release. You should begin using this reference in your recipes.

For more information, see *Source Metadata References*.

New math and statistical functions for arrays

The following functions can now be applied directly to arrays to derive meaningful statistics about them.

Function	Description
LISTSUM Function	Computes the sum of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type.
LISTMAX Function	Computes the maximum of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type.
LISTMIN Function	Computes the minimum of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type.

LISTAVERAGE Function	Computes the average of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type.
LISTVAR Function	Computes the variance of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type.
LISTSTDEV Function	Computes the standard deviation of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type.
LISTMODE Function	Computes the most common value of all numeric values found in input array. Input can be an array literal, a column of arrays, or a function returning an array. Input values must be of Integer or Decimal type.

Release 5.7

WEEKNUM function now behaves consistently across running environments

In Release 5.6 and earlier, the WEEKNUM function treated the first week of the year differently between the Trifacta Photon and Spark running environments:

- **Trifacta Photon week 1 of the year:** The week that contains January 1.
- **Spark week 1 of the year:** The week that contains at least four days in the specified year.

This issue was caused by Spark following an ISO-8601 standard and relying on the joda datetimeformatter.

Beginning in Release 5.7, the WEEKNUM function behaves consistently for both Trifacta Photon and Spark:

- **Week 1 of the year:** The week that contains January 1.

For more information, see *WEEKNUM Function*.

Release 5.6

URLPARAMS function returns null values

In Release 5.1 and earlier, the URLPARAMS function returned empty Objects when no answer was computed for the function.

In Release 5.6 and later, this function returns null values in the above case.

See *URLPARAMS Function*.

Release 5.1

Wrangle now supports nested expressions

Beginning in Release 5.1, all Wrangle functions now supported nested expressions, which can be arithmetic calculations, column references, or other function calls.

> **NOTE:** This feature is enabled by default, as this change does not break any steps created in previous versions of the product. It can be disabled if needed. See *Miscellaneous Configuration*.

> **NOTE:** This capability represents a powerful enhancement to the language, as you can now use dynamic inputs for all functions.

The following expression is a valid transform in Wrangle. It locates the substring in `myString` that begins with the `@` sign until the end of the string, inclusive:

```
derive value: substring(myString, find(myString, '@', true, 0), length(myString)
```

Nested arithmetic expressions:

Suppose you wanted just the value after the `@` sign until the end of the string. Prior to Release 5.1, the following generated a validation error:

```
derive value: substring(myString, find(myString, '@', true, 0) + 1, length(myString)
```

In the above, the addition of `+1` to the second parameter is a nested expression and was not supported. Instead, you had to use multiple steps to generate the string value.

Beginning in Release 5.1, the above single-step transform is supported.

Nested column references:

In addition to arithmetic expressions, you can nested column references. In the following example, the previous step has been modified to replace the static `+1` with a reference to a column containing the appropriate value (`at_sign_offset`):

```
derive value: substring(myString, find(myString, '@', true, 0) + at_sign_offset, length
(myString)
```

Nested function references:

Now, you can combine multiple function references into a single computation. The following computes the total volume of a cube of length `side` and then multiplies that volume by the number of cubes (`cube_count`) to compute the total `cube_volume`

```
derive type: single value: MULTIPLY(POW(cube_side,3),cube_count) as: 'cube_volume'
```

For more information, see *Wrangle Language*.

SOURCEROWNUMBER function generates null values consistently

The SOURCEROWNUMBER function returns the row number of the row as it appears in the original dataset. After some operations, such as unions, joins, and aggregations, this row information is no longer available.

In Release 5.0.1 and earlier, the results were confusing. When source row information was not available, the function was simply not available for use.

In Release 5.1 and later, the behavior of the SOURCEROWNUMBER function is more consistent:

- If the source row information is available, it is returned.
- If it is not available:
  - The function can still be used.
  - The function returns null values in all cases.

For more information, see *SOURCEROWNUMBER Function*.

New Functions

Function Name	Description
ARRAYELEMENTAT Function	Returns element value of input array for the provided index value.
DOUBLEMETAPHONE Function	Returns primary and secondary phonetic spellings of an input string using the Double Metaphone algorithm.
DOUBLEMETAPHONEEQUALS Function	Returns `true` if two strings match phonetic spellings using Double Metaphone algorithm. Tolerance threshold can be adjusted.
UNIQUE Function	Generates a new column containing an array of the unique values from a source column.

Release 5.0.1

RAND function generates true random numbers

In Release 5.0 and earlier, the RAND function produced the same set of random numbers within the browser, after browser refresh, and over subsequent runs of a job.

- During job execution, a default **seed value** was inserted as the basis for the function during the execution of the job.
- In some cases, this behavior is desired.

In Release 5.0.1 and later, the RAND function accepts an optional integer as a parameter. When this new seed value is inserted, the function generates deterministic, pseudo-random values.

- This version matches the behavior of the old function.

> **NOTE:** On all upgraded instances of the platform, references to the RAND function have been converted to use a default seed value, so that previous behavior is maintained in the upgraded version.

- If no seed value is inserted as a parameter, the RAND function generates true random values within the browser, after browser refresh, and over subsequent job runs.

> **NOTE:** Be aware that modifying your dataset based on the generated values of `RAND()` may have unpredictable effects later in your recipe and downstream of it.

For more information, see *RAND Function*.

Release 5.0

Required type parameter

Prior to Release 5.0, the following was a valid Wrangle step:

```
derive value:colA + colB as:'colC'
```

Beginning in Release 5.0, the `type` parameter is required. This parameter defines whether the transform is a single or multi-row formula. In the Transform Builder, this value must be specified.

The following is valid in Release 5.0:

```
derive type:single value:colA + colB as:'colC'
```

See *Derive Transform*.

See *Transform Builder*.

Deprecated aggregate transform

In Release 4.2.1 and earlier, the aggregate transform could be used to aggregate your datasets using aggregation functions and groupings.

In Release 5.0 and later, this transform has been merged into the pivot transform. The aggregate transform has been deprecated and is no longer available.

> **NOTE:** During upgrade to Release 5.0 and later, recipes that had previously used the aggregate transform are automatically migrated to use the pivot equivalent.

Example 1

Release 4.2.1 and earlier Aggregate:

```
aggregate value:AVERAGE(Scores)
```

Release 5.0 and later Pivot:

```
pivot value: AVERAGE(Score) limit: 1
```

The `limit` parameter defines the maximum number of columns that can be generated by the pivot.

Example 2

Aggregate:

```
aggregate value:AVERAGE(Scores) group:studentId
```

Pivot:

```
pivot group: StudentId value: AVERAGE(Score) limit: 1
```

For more information, see *Pivot Transform*.

New search terms

In the new Search panel, you can search for terms that can be used to select transformations for quick population of parameters. In the following table, you can see Wrangle how terminology has changed in Release 5.0 for some common transforms from earlier release.

> **Tip:** You can paste the Release 5.0 terms in the Search panel to locate the same transformations used in earlier releases.

Release 4.2.1 and earlier transforms	Release 5.0 and later search terms
aggregate	pivot

keep	filter
delete	filter
extract on:	extractpatterns
extract at:	extractpositions
extract before:	extractbetweendelimiters
extract after:	extractbetweendelimiters
replace on:	replacepatterns
replace at:	replacepositions
replace before:	replacebetweenpatterns
replace after:	replacebetweenpatterns
replace from:	replacebetweenpatterns
replace to:	replacebetweenpatterns
split on:	splitpatterns
split delimiters:	splitpositions
split every:	splitpositions
split positions:	splitpositions
split after:	splitpatterns
split before:	splitpatterns
split from:	splitpatterns
split to:	splitpatterns

Support for <> operator

Prior to Release 5.0, the following operator was used to test "not equal" comparisons:

```
! =
```

Beginning in Release 5.0, the following operators is also supported:

```
<>
```

Example:

```
derive value:IF ((col1 <> col2), 'different','equal') as:'testNotEqual'
```

> **Tip:** Both of the above operators are supported, although the <> operator is preferred.

For more information, see *Comparison Operators*.

ROUND function takes optional number of digits

The ROUND function now supports rounding to a specified number of digits. By default, values are rounded to the nearest integer, as before. See *ROUND Function*.

New Functions

Function Name	Description
DEGREES Function	Generates the value in degrees for an input radians value.
EXACT Function	Compares two strings to see if they are exact matches.
FILTEROBJECT Function	Filters the keys and values from an Object based on specified keys.
HOST Function	Returns the host value from a URL.
ISEVEN Function	Returns `true` if an Integer, function returning an Integer, or a column contains an even value.
ISODD Function	Returns `true` if an Integer, function returning an Integer, or a column contains an odd value.
KTHLARGESTUNIQUE Function	Computes the kth-ranked unique value in a set of values.
LCM Function	Returns the least common multiple between two input values.
MODE Function	Computes the mode (most common) value for a set of values.
MODEIF Function	Computes the mode based on a conditional test.
PAD Function	Pads the left or right side of a value with a specified character string.
PI Function	Generates the value for pi to 15 decimal places.
RADIANS Function	Generates the value in radians for an input degrees value.
RANDBETWEEN Function	Generates a random Integer in a range between two specified values.
RIGHTFIND Function	Locates a substring by searching from the right side of an input value.
ROLLINGCOUNTA Function	Computes count of non-null values across a rolling window within a column.
ROLLINGKTHLARGEST Function	Computes the *kth* largest value across a rolling window within a column.
ROLLINGKTHLARGESTUNIQUE Function	Computes the *kth* largest unique value across a rolling window within a column.
ROLLINGLIST Function	Computes list of all values across a rolling window within a column.
ROLLINGMAX Function	Computes maximum value across a rolling window within a column.
ROLLINGMIN Function	Computes minimum value across a rolling window within a column.
ROLLINGMODE Function	Computes mode (most common) value across a rolling window within a column.
ROLLINGSTDEV Function	Computes standard deviation across a rolling window within a column.
ROLLINGVAR Function	Computes variance across a rolling window within a column.
SIGN Function	Computes the positive or negative sign of an input value.
TRUNC Function	Truncates a value to the nearest integer or a specified number of digits.

URLPARAMS Function	Extracts any query parameters from a URL into an Object.
WEEKNUM Function	Calculates the week that the date appears during the year (1-52).

Release 4.2.1

None.

Release 4.2

New Filter transform

Perform a variety of predefined row filtrations using the new `filter` transform, or apply your own custom formula to keep or delete rows from your dataset.

- See *Remove Data*.
- See *Filter Transform*.

New Case transform

Beginning in Release 4.2, you can use the Transform Builder to simplify the construction of `CASE` statements. For each case, specify the conditional and resulting expression in separate textboxes.

- See *Apply Conditional Transformations*.
- See *Case Transform*.

Rename transform now supports multi-column rename

Use the `rename` transform to rename multiple columns in a single transform.

- See *Rename Columns*.
- See *Rename Transform*.

Drop specified columns or drop the others

The `drop` transform now supports the option of dropping all columns except the ones specified in the transform. See *Drop Transform*.

New string comparison functions

Compare two strings using Latin collation settings. See below.

NOW function returns 24-hour time values

In Release 4.1.1 and earlier, the NOW function returned time values for the specified time zone in 12-hour time, which was confusing.

In Release 4.2 and later, this function returns values in 24-hour time.

New Transforms

Transform Name	Documentation
case	*Case Transform*
filter	*Filter Transform*

New Functions

Function Name	Documentation
STRINGGREATERTHAN	*STRINGGREATERTHAN Function*
STRINGGREATERTHANEQUAL	*STRINGGREATERTHANEQUAL Function*
STRINGLESSTHAN	*STRINGLESSTHAN Function*
STRINGLESSTHANEQUAL	*STRINGLESSTHANEQUAL Function*
SUBSTITUTE	*SUBSTITUTE Function*

Release 4.1

Standardization page and transform have been removed

For a number of releases, the Standardization page and its related transform have been available via feature flag, due to a number of issues.

> **NOTE:** This feature was not typically enabled in Trifacta Wrangler Enterprise deployments.

This feature has been removed from the product altogether and will be replaced in the future by a more robust standardization and normalization capability.

Syntax Changes

New Transforms

Transform Name	Documentation	Notes
comment	*Comment Transform*	In previous releases, you could insert comments of the following format: `// This is a comment.` Beginning in Release 4.1, comments are supported by formal transform.

New Functions

Function Name	Documentation
CASE	*CASE Function*

Release 4.0.1

Map data type is now Object data type

The Map data type has been renamed to the Object data type. There are no changes to the behavior.

> **NOTE:** This change is not reflected in the Release 4.0.1 PDF documentation.

Splitrows transform now permits specifying of quote escaping character

For text-based formats that format fields between quotes, you can specify the character that is used to signify the escaping of the quote character in the data. The `quoteEscapeChar` parameter identifies the character in the data the precedes quotes that are supposed to be part of the data, instead of the marker for a field. See *Splitrows Transform*.

Release 4.0

Script steps displayed in natural language

In Release 3.2.1 and earlier, the steps of your recipe were displayed in raw form of Wrangle, as in the following example:

```
split col: column1 on: ',' limit: 5 quote: '\"'
```

Beginning in Release 4.0, by default, recipe steps are displayed in a more natural form of language, so that you can read the intention of the step without having to understand the details of the underlying language syntax. In natural language format, the above step is rendered as the following:

```
Split column1 on ',' 5 times
```

Notes:

- If you edit a natural language version of your step, you perform your edits in Wrangle.
- If preferred, you can switch back to displaying in source Wrangle. In the data grid, click the Data Grid options button. Select **Show Wrangle Script**. See *Data Grid Panel*.
- Recipe steps are listed in the product documentation in source Wrangle, so that you can copy and paste them into the Transform Builder as needed.

`set`, `derive`, and `window` transforms can now perform any type of computation

To support the above capabilities, the following changes appear in the language:

- `derive` transform now supports `window` functions. See *Derive Transform*.
    - See *Window Functions*.
- `set` and `window` transforms now support `aggregate` functions.
    - See *Set Transform*.
    - See *Window Transform*.
    - See *Aggregate Functions*.

Multi-column input support

- `set` and `settype` transforms now support multiple input columns.
- When working with multiple columns, `set` transform now accepts a placeholder variable in the formula.
- See *Set Transform*.
- See *Settype Transform*.

Syntax Changes

Terminology Changes

`rollingaverage` **function accepts two windowing parameters.**

In Release 3.2.1 and earlier, you could specify your window for computing the rolling average using a single parameter.

```
window value: ROLLINGAVG(POS_Sales, 3) order: Whse_Nbr
```

This single parameter determined the row offset after the current row. The above transform captures a window of values from the current row forward two rows for the rolling average value. There was no way to capture a window that included values that were both before and after the current row.

Beginning in Release 4.0, the function accepts an additional parameter, which enables computation across a before/after window. The following example computes the rolling average from two rows before and two rows after the current row:

```
window value: ROLLINGAVERAGE(POS_Sales, 3, 2) order: Whse_Nbr
```

Notes:

- The function name has changed to ROLLINGAVERAGE.
- The behavior of the first parameter has changed. It captures rows before the current one, instead of rows after the current one.
  - The default values are -1 and 0, which capture all values from the current row back to the first row of the dataset.
- During the upgrade process, transform steps using this function are automatically migrated to the new method of specification.
- For more information, see *ROLLINGAVERAGE Function*.

Function Changes

The following name changes have been applied to existing functions to use more familiar names.

Old Function Name	New Function Name	Notes
ROLLINGAVG	ROLLINGAVERAGE	

New Datetime functions

These functions generate date and timestamps at execution time:

Function Name	Description
NOW	See *NOW Function*.
TODAY	See *TODAY Function*.
DATETIME	See *DATETIME Function*.

New conditional functions

These functions are conditionals based on data validation against a column's data type:

Function Name	Description

IFMISSING	See *IFMISSING Function.*
IFNULL	See *IFNULL Function.*
IFVALID	See *IFVALID Function.*
IFMISMATCHED	See *IFMISMATCHED Function.*

These functions compute specific values based on conditionals:

Function Name	Description
ANYIF	See *ANYIF Function.*
AVERAGEIF	See *AVERAGEIF Function .*
COUNTAIF	See *COUNTAIF Function .*
COUNTDISTINCTIF	See *COUNTDISTINCTIF Function.*
COUNTIF	See *COUNTIF Function.*
KTHLARGESTIF	See *KTHLARGESTIF Function.*
LISTIF	See *.LISTIF Function.*
MAXIF	See *MAXIF Function.*
MINIF	See *MINIF Function.*
STDEVIF	See *STDEVIF Function.*
SUMIF	See *SUMIF Function.*
VARIF	See *VARIF Function.*

Other new functions

Function Name	Description
COUNTA	See *COUNTA Function.*
ROLLINGSUM	See *ROLLINGSUM Function.*
ROWNUMBER	See *ROWNUMBER Function.*
SUFFIX	See *SUFFIX Function.*
STARTSWITH	See *STARTSWITH Function.*
ENDSWITH	See *ENDSWITH Function.*

`set` transform no longer accepts `row` parameter

In Release 3.2.1 and earlier, the `row` parameter could be used to filter the rows in a dataset to which the `set` transform value is applied, as in the following example:

```
set col: results value: 'Outstanding!' row: (score == 100)
```

Beginning in Release 4.0, the `row` parameter has been removed. Instead, you can specify conditionals in the `value` parameter. During upgrade, the above transform step is converted to the following:

```
set col: results value: IF(score == 100, 'Outstanding!', '')
```

In addition to the standard IF function, you can apply any of the new conditional functions listed below.

For more information, see *Set Transform*.

Format string with # before 0 is no longer supported in the NUMFORMAT function

In Release 3.2.1, the NUMFORMAT function supported a format string of ##.#0 in the Javascript running environment. This string was not supported in the Trifacta Photon running environment.

For Release 4.0 and later, this format string is no longer supported and must be changed.

> **NOTE:** After you have upgraded to Release 4.0 or later, you must change references format strings with a # before 0 for the NUMFORMAT function to use a supported formatting string. See *NUMFORMAT Function*.

ARRAYUNIQUE function can now take a single column as input

In Release 3.2.1 and earlier, the ARRAYUNIQUE function required at least two functions to generate an output.

Beginning in Release 4.0, this function can accept a single array or column as input, generating an output array containing only the unique values in the source. See *ARRAYUNIQUE Function*.

Execution Changes

Ternary predicates evaluating to null return false expressions

Suppose you have the following function expression:

```
IF(NULL(),1,2)
```

In Release 3.2.1 and earlier, predicates that returned null values returned null for the entire expression. In this case, the expression returned a null value.

In Release 4.0 and later, this expression returns 2.

See *IF Function*.

Null values no longer automatically filtered in limiting transforms

In Release 3.2.1 and earlier, when filtering the set of rows using a recipe step, such as a keep or delete transform, any null values in the evaluated in the condition would result in the filtering being applied. Example:

```
delete row:invAge >=90
```

If invAge contained a null value for a row, the row was deleted.

> **NOTE:** In Release 4.0 and later, null values used as inputs to filtering transforms do not result in the row being filtered. This is a change in behavior for null values.

For each of the transforms below, you can review how to retain the Release 3.2.1 and earlier behavior in Release 4.0 and later.

Delete transform:

Release 3.2.1 example:

```
delete row:invAge >=90
```

Release 4.0 example:

```
delete row:(invAge >=90 && invAge == null())
```

Keep transform:

Release 3.2.1 example:

```
keep row:POS_Sales < 100
```

Release 4.0 example:

```
keep row:(POS_Sales < 100 && POS_Sales != null())
```

IF transform:

Release 3.2.1 example:

```
derive value:IF(rating > 9.0, 'ok','retry') as:'status'
```

Release 4.0 example:

```
derive value:(IF(rating > 9.0, 'ok','retry') && rating != null()) as:'status'
```

> **Tip:** Release 4.0 introduces a series of conditional functions that can streamline computation and action. These functions test conditionals based on type (e.g. `IFNULL`) or based on computation of an aggregate function (e.g. `SUMIF`). See *New conditional functions* above.

For more information:

- *Delete Transform*
- *Keep Transform*
- *IF Function*

Release 3.2.1

Syntax Changes

Terminology Changes

> **NOTE:** Beginning in Release 3.2.1, values that are considered `empty` are now referred to as `missing`.

Function Changes

The following name changes have been applied to existing functions to use more familiar names.

Old Function Name	New Function Name	Notes
MEAN	AVERAGE	
CEIL	CEILING	
DATEDIFF	DATEDIF	
DAYOFWEEK	WEEKDAY	
UPPERCASE	UPPER	
LOWERCASE	LOWER	
PROPERCASE	PROPER	
LENGTH	LEN	
ARRAYLENGTH	ARRAYLEN	
CONCAT	ARRAYCONCAT	
CROSS	ARRAYCROSS	
INTERSECTION	ARRAYINTERSECT	
KEYSET	KEYS	
UNIQUE	ARRAYUNIQUE	
ZIP	ARRAYZIP	
EMPTY	ISMISSING	
ISEMPTY	ISMISSING	Removal of duplicate function name.
MISMATCHED	ISMISMATCHED	
VALID	ISVALID	
WINDOWFILL	FILL	
MODULO	MOD	

Execution Changes

Transforms that nest null values in arrays now write `null` literals on Trifacta Photon

In Release 3.2 and earlier, when a transform step was nested a null value within an array, an empty string value was written in the Trifacta Photon running environment.

In Release 3.2.1 and later, the value written for a nested null value in the array is the literal: `null`.

Suppose your data looks like the following:

text_col	empty_str_col	null_col
myText		

where:

- `empty_str_col` contains an empty string value.
- `null_col` contains a null value

If you add the following recipe step:

```
nest col: text_col, empty_str_col, null_col into: array as: 'result'
```

In Release 3.1, the result was the following:

text_col	empty_str_col	null_col	result
myText			["myText", "", ""]

In Release 3.2.1, the result is the following:

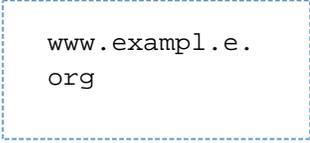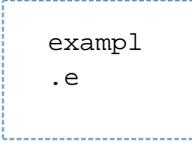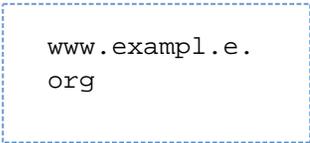| text_col	empty_str_col	null_col	result
myText			["myText", "", null]

This change was made to align the behaviors of the Trifacta Photon running environment with the JavaScript running environment.

Release 3.2

Syntax Changes

Transform Changes

- `multisplit` transform has been deprecated. All `multisplit` capabilities are now supported by the `spl it` transform. See *Split Transform*.
- `pivot` transform now supports multiple columns. See *Pivot Transform*.
- `unnest` transform now requires the `keys` parameter, which was optional in previous releases. See *Unnest Transform*.
    - To unnest arrays without specifying keys, use the `flatten` transform. See *Flatten Transform*.
- `arraylength` and `arraystomap` now accept functions that return arrays as inputs to the function.
    - See *ARRAYLEN Function*.
    - See *ARRAYSTOMAP Function*.
- `domain` and `subdomain` functions have been updated to reflect standard interpretations of domain and sub-domain values for URLs:

Release	Example URL	Domain Function	Subdomain Function
Release 3.1.2 and earlier	`www.exampl.e. org`	`exampl .e`	`www`
Release 3.2 and later	`www.exampl.e. org`	`e`	`www. exampl`
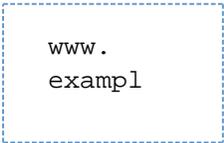
- See *DOMAIN Function*.
- See *SUBDOMAIN Function*.

- The following parameter values are no longer supported as special capture groups in the `with` parameter for the `replace` transform. These references in the `with` parameter do not work at scale in any supported running environment:

```
$&
$`
$'
```

Parameter Changes

Applicable Transform(s)	Old Term	New Term	Notes
Extract Transform	urlparam		Removed from use in the `extract` transform. Use of this parameter in that transform prevented the use of other `extract` parameters. See *Extract Transform*.
Countpattern Transform	quote		Removed from use in the `countpattern` transform. Parameter was not being respected. See *Countpattern Transform*.
Extractkv Transform	quote		Remove from use in the `extractkv` transform. Parameter was not being used. See *Extractkv Transform*.
Extractlist Transform	quote		Parameter is now used exclusively for matching against delimiters. Parameter does not match against patterned values. See *Extractlist Transform*.
Split Transform and *Extract Transform*	limit		For these transforms, the `limit` parameter can no longer be used in conjunction with the following parameters: `positions`, `delimiters`, `at`, and `urlparam`. In previous releases, these combinations did not actually work, even though the transform step was consumed. Now, it generates an error.

Function Changes

More consistent results for DATEDIFF functions:

Prior to Release 3.2, the DATEDIFF function generated inconsistent results between the Pig and Javascript running environments for DayOfYear calculations.

Beginning in Release 3.2, the DATEDIFF function has been updated to generate more consistent results. See *DATEDIF Function*.

Trifacta Pattern Changes

Changes to alpha-numeric pattern:

The `alpha-numeric` Trifacta pattern now applies to a single character and does not match on underscores (_). Previously, it was applied to one or more alpha-numeric characters, as well as underscores.

> **NOTE:** Beginning in Release 3.2, the `alpha-numeric` Trifacta pattern applies to a single character. If you used it in your recipes prior to Release 3.2, these references have been converted to regular expressions to support matching with multiple characters.

New alphanum-underscore pattern:

To support previous functionality, you can use the new `alphanum-underscore` pattern, which matches on a single alpha-numeric character or underscore.

For more information, see *Text Matching*.

RANGE function

The `RANGE` function now accepts negative start and stop values.

Deprecated Items

Aggregate Tool

In prior releases, you could build aggregation steps using a separate tool, which was available through the Transform Editor.

In Release 3.2, this tool has been replaced by building `aggregate` transforms in the Transform Builder. See *Transform Builder*.

Language Cheat Sheet

You can still access the Language Cheat Sheet by adding `/docs` to the base URL. For example:

```
http://www.example.com:3005/docs
```

> **NOTE:** The above option is likely to be deprecated in a future release.

Changes to the APIs

Contents:

- *Changes for Release 6.0*
  - *Error in Release 6.0.x API docs*
  - *Planned End of Life of v3 API endpoints*
- *Changes for Release 5.9*
  - *Introducing Access Tokens*
- *Changes for Release 5.1*
- *Changes for Release 5.0*
  - *Introducing v4 APIs*
- *Changes for Release 4.2*
  - *Create Hive and Redshift connections via API*
  - *WrangledDataset endpoints are still valid*

Review the changes to the publicly available REST APIs for the Trifacta® platform for the current release and past releases.

Changes for Release 6.0

Error in Release 6.0.x API docs

In Release 6.0 - Release 6.0.2, the online and PDF versions of the documentation referenced the following endpoint: API JobGroups Get Status v4. According to the docs, this endpoint was triggered in this manner:

Method	GET
Endpoint	/v4/jobGroups/<id>/status

This endpoint exists in v3 of the API endpoints. It does not exist in v4.

Instead, you should monitor the `status` field for the base GET endpoint for jobGroups. For more information, see *API JobGroups Get v4*.

Planned End of Life of v3 API endpoints

In Release 6.0, the v3 API endpoints are supported.

> **In the next release of Trifacta Wrangler Enterprise after Release 6.0, the v3 API endpoints will be removed from the product (End of Life).**

You must migrate to using the v4 API endpoints before upgrading to the next release after Release 6.0. For more information, see *API Migration to v4*.

Changes for Release 5.9

Introducing Access Tokens

Each request to the API endpoints of the Trifacta platform requires submission of authentication information. In Release 5.1 and earlier:

- A request could include clear-text username/password combinations. This method is not secure.
- A request could include a browser cookie. This method does not work for well for use cases outside of the browser (e.g. scripts).

Beginning in Release 5.9, API users can manage authentication using access tokens. These tokens obscure any personally identifiable information and represent a standards-based method of secure authentication.

> **NOTE:** All previous methods of API authentication are supported in this release. Access tokens is the preferred method of authentication.

The basic process works in the following manner:

1. API user requests generation of a new token.
    1. This initial request must contain a valid username and password.

      2. Request includes expiration.

      3. Token value is returned in the response.

2. The token value inserted into the Authorization header of each request to the platform.
3. User monitors current time and expiration time of the token. At any time, the user can request a new token to be generated using the same endpoint used in the initial request.

Access tokens can be generated via API or the Trifacta application.

> **NOTE:** This feature must be enabled in your instance of the platform. See *Enable API Access Tokens*.

- **API:** For more information, see *API AccessTokens Create v4*.
- **Trifacta application:** For more information, see *Access Tokens Page*.

For more information on API authentication methods, see *API Authentication*.

Changes for Release 5.1

None.

Changes for Release 5.0

Introducing v4 APIs

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

Release 5.0 signals the introduction of version 4 of the REST APIs.

> **NOTE:** At this time, a very limited number of v4 REST APIs are publicly available. Where possible, you should continue to use the v3 endpoints. For more information, see *v4 Endpoints*.

> **NOTE:** v3 of the APIs are still supported as of the generally available date for Release 5.0. For more information, see *v3 Endpoints*.

v4 conventions

The following conventions apply to v4 and later versions of the APIs:

- Parameter lists are consistently enveloped in the following manner:

```
{ "data": [
    {
        ...
    }
  ]
}
```

- Field names are in `camelCase` and are consistent with the resource name in the URL or with the `embed` U
RL parameter.

- From early API versions, foreign keys have been replaced with identifiers like the following:

v3 and earlier	v4 and later

`"createdBy": 1,`	`"creator": {`
	`  "id": 1`
	`},`

`"updatedBy": 2,`	`"updater": {`
	`  "id": 2`
	`},`

- Publication endpoint references database differently. This change is to make the publishing endpoint for relational targets more flexible in the future.

v3 and earlier	v4 and later
`"database": "dbName",`	`"path": ["dbName"],`

Changes for Release 4.2

Create Hive and Redshift connections via API

You can create connections of these types via API:

- Only one global Hive connection is still supported.
- You can create multiple Redshift connections.

See *API Connections Create v3*.

WrangledDataset endpoints are still valid

In Release 4.1.1 and earlier, the WrangledDataset endpoints enabled creation, modification, and deletion of a wrangled dataset object, which also created the associated recipe object.

In Release 4.2, wrangled datasets have been removed from the application interface. However, the WrangledDataset API endpoints remain, although they now apply to the recipe directly.

The following endpoints and methods are still available:

> **NOTE:** In a future release, these endpoints may be migrated to recipe-based endpoints. API users should review this page for each release.

- *API WrangledDatasets Create v3*
- *API WrangledDatasets Get List v3*

- *API WrangledDatasets Get v3*
- *API WrangledDatasets Delete v3*
- *API WrangledDatasets Get PrimaryInputDataset v3*
- *API WrangledDatasets Put PrimaryInputDataset v3*

For more information, see *Changes to the Object Model*.

Changes to Configuration

Contents:

- *Release Updates*
  - *Release 6.0*
- *Configuration Mapping*

To centralize common enablement configuration on a per-workspace basis, a number of configuration properties are being migrated from `trifacta-conf.json` into the Trifacta® application.

- In prior releases, some of these settings may have been surfaced in the Admin Settings page. See *Admin Settings Page*.
- For more information on configuration in general, see *Platform Configuration Methods*.

To assist administrators in managing these settings, these section provides a per-release set of updates and a map of old properties to new settings.

- These settings now appear in the Workspace Admin page. To access, from the left menu bar select **Settin gs menu > Settings > Workspace Admin**.
- For more information, see *Workspace Admin Page*.

Release Updates

Release 6.0

Initial release of the Workspace Admin page. See below for configuration mapping.

Configuration Mapping

The following mapping between old `trifacta-conf.json` settings and new Workspace Admin settings is accurate for the current release.

Admin Settings or `trifacta-conf.json` setting	Workspace Admin setting
webapp.walkthrough.enabled	Product walkthroughs
webapp.session.durationInMins	Session duration
webapp.enableDataDownload	Sample downloads
webapp.enableUserPathModification	Allow the user to modify their paths
webapp.enableSelfServicePasswordReset	Enable self service password reset
outputFormats.Parquet	Parquet output format
outputFormats.JSON	JSON output format

outputFormats.CSV	CSV output format
outputFormats.Avro	Avro output format
outputFormats.TDE	TDE output format
feature.scheduling.enabled	Enable Scheduling feature
feature.scheduling.schedulingManagementUI	Scheduling management UI (experimental)
feature.scheduling.upgradeUI	Show users a modal to upgrade to a plan with Scheduling
feature.sendFlowToOtherPeople.enabled	Allow users to send copies of flows to other users
feature.flowSharing.enabled	Enable Flow Sharing feature
feature.flowSharing.upgradeUI	Show users a modal to upgrade to a plan with Flow Sharing
feature.enableFlowExport	Allow users to export their flows
feature.enableFlowImport	Allow users to import flows into the platform
feature.hideAddPublishingActionOptions	Forbid users to add non-default publishing actions
feature.hideUnderlyingFileSystem	Hide underlying file system to users
feature.publishEnabled	Enable publishing
feature.rangeJoin.enabled	Enable UI for range join
feature.showDatasourceTabs	Show datasource tabs in the application
feature.showFileLocation	Show file location
feature.showOutputHomeDirectory	Show output directory in profile view
feature.showUploadDirectory	Show upload directory in profile view
metadata.branding	Branding to use in-product for this deployment
webapp.connectivity.enabled	Enable Connectivity feature
webapp.connectivity.upgradeUI	Show users a modal to upgrade to a plan with Connectivity
feature.apiAccessTokens.enabled	API Access Token

Changes to the Command Line Interface

Contents:

- *Changes for Release 6.0*
- *Changes for Release 5.1*
- *Changes for Release 5.0*
  - *CLI for Connections does not support Redshift and SQL DW connections*
- *Changes for Release 4.2*
  - *All CLI scripts with relational connections must be redownloaded*
  - *Redshift credentials format has changed*
- *Changes for Release 4.1.1*
  - *Single-file CLI publishing method is deprecated*
  - *CLI run job output has changed*
- *Changes for Release 4.1*
  - *CLI run job output has changed*
  - *Specify multiple publication targets for a job through an external file*
  - *Create schema command type is no longer available*
  - *Pig job type is no longer available*
  - *Default CLI job type is now Photon*
- *Changes for Release 4.0*
  - *JS job type is no longer supported*

- *Pig job type is being replaced*
- *Changes to the Hive Connection*
- *Changes for Release 3.2.1*
  - *New file publishing actions and parameters*
  - *Change in behavior for output_path parameter*
  - *CLI commands now accept only a single output format*
- *Changes for Release 3.2*
  - *CLI supports Kerberos-based credentials*
  - *Transfer assets of a deleted user to another user*
- *Changes for Release 3.1.2*
  - *cli_output_path requires additional permissions*
  - *Changes to command outputs and responses for connections*
- *Changes for Release 3.1*
  - *CLI Tools Available as Executables*
  - *Download Upgraded Transform Scripts*
  - *Behavior Changes*
  - *Parameter Changes*
  - *Example Commands*
- *Changes for Release 3.1.1*
  - *Improved error messaging*
  - *Update to Hive params file*
  - *Changes to CLI for Connections*
  - *Changes to connection example commands*
  - *Updates to connections documentation*

When you upgrade to a new version of Trifacta Wrangler Enterprise, you must download your CLI packages from the Transformer page after the upgrade is complete. You may be able to execute your packages from the previous release, but backward compatibility of exported CLI packages is not explicitly supported.

Changes for Release 6.0

In the next release of Trifacta Wrangler Enterprise after Release 6.0, the Command Line Interface (CLI) will reach its End of Life (EOL). The tools will no longer be included in the software distribution and will not be supported for use against the software platform. You should transition away from using the CLI as soon as possible. For more information, see *CLI Migration to APIs*.

Changes for Release 5.1

None.

Changes for Release 5.0

CLI for Connections does not support Redshift and SQL DW connections

In Release 5.0, the management of Redshift and SQL DW connections through the CLI for Connections is not supported.

NOTE: Please create Redshift and SQL DW connections through the application. See *Connections Page*.

Changes for Release 4.2

All CLI scripts with relational connections must be redownloaded

Each CLI script that references a dataset through a connection to run a job must be re-downloaded from the application in Release 4.2.

Scripts from Release 4.1 that utilize the `run_job` command do not work in Release 4.2.

Requirements for Release 4.2 and later:

1. In the executing environment for the CLI script, the relational (JDBC) connection must exist and must be accessible to the user running the job.
2. When the CLI script is downloaded from the application, the connection ID in the `datasources.tsv` must be replaced by a corresponding connection ID from the new environment.
    1. Connection identifiers can be retrieved using the `list_connections` command from the CLI. See *CLI for Connections*.

After the above changes have been applied to the CLI script, it should work as expected in Release 4.2. For more information, see **Run job** in *CLI for Jobs*.

Redshift credentials format has changed

In Release 4.1.1 and earlier, the credentials file used for Redshift connection was similar to the following:

```
{
 "awsAccessKeyId": "<your_awsAccessKeyId>",
 "awsSecretAccessKey": "<your_awsSecretAccessKey>",
 "user": "<your_user>",
 "password": "<your_password>"
}
```

In Release 4.2:

- The AWS key and secret, which were stored in `trifacta-conf.json`, do not need to be replicated in the Redshift credentials file.
- The Trifacta platform now supports EC2 role-based instance authentication. This configuration can be optionally included in the credentials file.

The credentials file format looks like the following:

```
{
  "user": "<your_user>",
  "password": "<your_password>",
  "iamRoleArn": "<your_IAM_role_ARN>"
}
```

> **NOTE:** For security purposes, you may wish to remove the AWS key/secret information from the Redshift credentials file.

> **NOTE:** `iamRoleArn` is optional. For more information on using IAM roles, see
> *Configure for EC2 Role-Based Authentication*.

Changes for Release 4.1.1

Single-file CLI publishing method is deprecated

In Release 4.1 and earlier, the `run_job` action for the Command Line Interface supported specifying a single-file publishing action as part of the command itself.

In Release 4.1.1 and later, this method has been superseded by specifying single- and multi-file publishing actions through an external file. While the single-file version is likely to still work with the platform, it is likely to be removed from the platform in the future.

See *CLI Publishing Options File*.

CLI run job output has changed

> **NOTE:** This change first appeared in Release 4.1 but was not surfaced in documentation until Release 4.1.1.

See below.

Changes for Release 4.1

CLI run job output has changed

In Release 4.0 and earlier, standard output for launching a job is similar to the following:

```
Job has been successfully launched:
You may monitor the progress of your job here: http://localhost:3005/jobs
Upon success, you may view the results of your job here:
http://localhost:3005/jobs/34
```

In Release 4.1 and later, standard output has been changed to the following:

```
Job #34 has been successfully launched:
You may monitor the progress of your job here: http://localhost:3005/jobs
```

> **NOTE:** If your CLI script processing relies on the standard output for gathering job identifiers, you must modify how you collect the jobId to match the new format. For an example of how to parse the standard output to gather jobId's, see *CLI Example - Parameterize Job Runs*.

For more information, see *CLI for Jobs*.

Specify multiple publication targets for a job through an external file

In Release 4.0 and earlier, a `run_job` command could generate a single publication.

In Release 4.1 and later, you can specify multiple publication targets to be executed when a job is run. These publication targets are specified in an external file.

> **NOTE:** The legacy method of defining output options are part of individual parameters in the `run_job` command is still supported. However, it is likely to be deprecated in a future release. You should migrate your CLI scripts to use the new method that references external publication files.

Release 4.0 Example:

```
/trifacta_cli.py run_job --user_name <trifacta_user> --password
<trifacta_password> --job_type spark
--output_format csv --data redshift-test/datasources.tsv --script
redshift-test/script.cli
--publish_action create --header true --single_file true
--cli_output_path ./job_info.out --profiler on --output_path
hdfs://localhost:8020/trifacta/queryResults/foo@trifacta.com/MyDataset/43
/cleaned_table_1.csv
```

Release 4.1 Example:

Following references the `--publish_opt_file` parameter.

```
./trifacta_cli.py run_job --user_name <trifacta_user> --password
<trifacta_password> --job_type spark --data redshift-test/datasources.
tsv --script redshift-test/script.cli --cli_output_path ./job_info.out --
profiler on --publish_opt_file /json/publish/file/publishopts.json
```

The following parameters have been migrated into settings in the publishing options file:

- `output_format`
- `publish_action`
- `header`
- `single_file`
- `output_path`

For more information, see *CLI Publishing Options File*.

Create schema command type is no longer available

In Release 4.0 and earlier, you could use the `create_schema` command to create an empty schematized table in Hive or Redshift.

In Release 4.1 and later, this command is no longer available. With ad-hoc publication to Hive now available, you shouldn't need to use this command.

Pig job type is no longer available

In Release 4.0 and earlier, you could specify `-job_type pig` to execute a job on the Hadoop Pig running environment.

In Release 4.1, the Hadoop Pig running environment has been deprecated.

> **NOTE:** CLI scripts that reference the `pig` job type must be updated.

1. Search your CLI scripts for:

   ```
   pig
   ```

2. For references to this type of job, replace with one of the following references, depending on your deployment:

Job Type Value	Description
`spark`	Runs job in the Spark running environment.
`hadoop`	Runs jobs in the default running environment for the Hadoop cluster. For this release, that environment is Spark. This setting future-proofs against subsequent changes to the default Hadoop running environment.
`photon`	Runs job on the Trifacta node. This setting is only recommended for smaller jobs.

3. Save your scripts.
4. Run an example of each on your upgraded Release 4.1 instance.

Default CLI job type is now Photon

In Release 4.0 and earlier, the default job type for CLI jobs was `JS`. If the execution engine was not specified, then the job was run on the Javascript-based execution engine.

In Release 4.1 and later, the Javascript execution engine has been deprecated, and the default job type for CLI jobs is now `Photon`, which is the default execution engine on the Trifacta node.

> **NOTE:** If your CLI scripts do not specify a `job_type` parameter, the job is executed on the Photon running environment, which replaces the Javascript running environment. If this is acceptable, no action is required.
>
> Otherwise, you must review your scripts and manually specify a `job_type` parameter for execution.

For more information, see *CLI for Jobs*.

Changes for Release 4.0

JS job type is no longer supported

Beginning in Release 4.0, the Javascript running environment is no longer supported.

In Release 4.0, you may be able to continue to use this running environment to execute your jobs. However, there may be differences in terms of functionality between CLI jobs executed under this running environment and the results generated through the GUI.

After upgrade, you should review all of your scripts for their `--job_type` values. You should change these references to the following:

```
--job_type photon
```

For more information, see *CLI for Jobs*.

Pig job type is being replaced

Beginning in Release 4.0, the Pig running environment is being superseded by the new Spark running environment. The Spark running environment for execution is the default for Release 4.0 and later.

For CLI jobs, you should modify your script to use the following job type:

```
--job_type hadoop
```

The above option instructs the CLI to execute the job on the default running environment for the integrated Hadoop cluster.

- For upgrading customers who have not enabled Spark, the job is executed on Pig, as in previous releases.
- In a future release, when the Pig running environment is deprecated, this setting will still apply to the default running environment (Spark), and your scripts will not need to be updated.

For more information, see *CLI for Jobs*.

Changes to the Hive Connection

> **NOTE:** If you are upgrading from a previous version in which the Trifacta node is connected to Hive, you must recreate the Hive connection through the Command Line Interface.

In Release 3.2 and 3.2.1, in some environments, the Hive connection was created using a `trifacta-conf.json` value that was deprecated. The configuration looked like the following:

```
"deprecated": {   "kerberos": {
    "principals": {
      ...
        "hive": "hive/_HOST@EXAMPLE.COM"
      ...
    },
  },
}
```

In Release 4.0 and later, this issue has been fixed by moving this principal value into the connection string options for the parameters file for the connection:

```
    "connectStrOpts": ";principal=<principal_value>",
```

For more information, see *Configure for Hive*.

Changes for Release 3.2.1

New file publishing actions and parameters

Beginning in Release 3.2.1, you can specify file publishing options through the command line interface:

Option	Description
create, append, or overwrite	In the `publish_action` parameter, you specify whether you want new executions of the command to create a new file, append the new results to the existing file, or overwrite any file found in the specified location. These options match the new file publishing options available through the application in Release 3.2.1.
header	If the `header` flag is included, the column headers are added as the first row in any CSV output.
single_file	If the `single_file` flag is included, the output is written as a single file. Otherwise, the output is written as a part file within the given path.

For more information, see *CLI for Jobs*.

Change in behavior for `output_path` parameter

Prior to Release 3.2.1:

The `output_path` parameter was optional for inclusion in a `run_job` command.

It was used to specify a path to a folder. This folder was populated with the output files from runs of the `run_job` command. The filename was generated by the CLI.

If this parameter was not specified, outputs were written to the default output location specified for the executing user.

Release 3.2.1 and later:

This parameter must be included in `run_job` commands.

In Release 3.2.1 and later, this parameter now specifies a fully qualified URI to the output file, such as the following examples:

```
hdfs://host:port/path/filename.csv
s3://bucketName/path/filename.csv
```

> **NOTE:** If you created scripts that use the CLI for releases before Release 3.2.1, you must update any `run_job` commands to include the `output_path` parameter.

Depending on the `publish_action` parameter, the following behaviors are applied to the file on subsequent job executions:

Publish action	Description
`create`	A new file is created with each job: `filename_1.csv`, `filename_2.csv`, etc.
`append`	Any existing file is appended with the results from the new job execution.
`overwrite`	Any existing file is replaced with the results from the new job execution.

For more information, see *CLI for Jobs*.

CLI commands now accept only a single output format

Prior to Release 3.2.1, CLI commands could deliver results to multiple formats in a single command.

In Release 3.2.1, a CLI command can now accept only a single output format, since this output is now associated with a single publishing action, publishing options, and `output_path`. Additionally, the `output_formats` parameter has been changed to `output_format`.

> **NOTE:** Any CLI command from a release prior to Release 3.2.1 that specifies an `output_formats` parameter must be updated to use the new parameter name: `output_format`.

See *CLI for Jobs*.

Changes for Release 3.2

CLI supports Kerberos-based credentials

Beginning in Release 3.2, the Trifacta command line interface:r=true supports the use of Kerberos-based credentials for interactions with the Trifacta platform and Hadoop backend datastore.

Steps:

1. To enable use of Kerberos, see *Configure for Kerberos Integration*.
2. In particular, you must configure a Kerberos principal for the Trifacta platform to use.
3. Then, you must enable use of the Kerberos keytab by the CLI.

See the above link for further instructions.

After you have performed the above configuration, you do not need to provide a password value for commands issued through the Trifacta CLI. See *CLI for Jobs*.

Transfer assets of a deleted user to another user

When you delete a user through the CLI, you can now transfer that user's assets to a new user. Add the `--transfer_assets_to` parameter, which is followed by the userId of the user who is now the owner of them.

Example command:

```
./trifacta_admin_cli.py --admin_username <trifacta_admin_user> --
admin_password <trifacta_admin_password> delete_user --user_name
joe@example.com --transfer_assets_to jim@example.com
```

For more information, see *CLI for User Admin*.

Changes for Release 3.1.2

cli_output_path requires additional permissions

In any command that utilizes a `cli_output_path` parameter, the user who executes the command must also have exec permissions on all parent folders to the `cli_output_path` folder.

- This issue affects Release 3.1.1 and later.

Changes to command outputs and responses for connections

In Release 3.1.2, the outputs and responses for connection-related commands in the CLI have changed. Differences are listed below.

For more information, see *CLI for Connections*.

Create connection

Example command (same for both releases):

```
./trifacta_cli.py create_connection --user_name <trifacta_user> --
password <trifacta_password>
--conn_type microsoft_sqlserver --conn_name aSQLServerConnection
--conn_description "This is my connection."
--conn_host example.com --conn_port 1234
--conn_credential_type basic
--conn_credential_location ~/.trifacta/config_conn.json
--conn_params_location ~/.trifacta/p.json
--cli_output_path ./conn_create.out
```

Release 3.1.1:

Output:

```
Creating connection aSQLServerConnection
Connection information for aSQLServerConnection
  description: This is my connection.
  host: example.com
  credentials: ["{u'username': u'<trifacta_user>'}"]
  port: 1234
  is_global: False
  name: aSQLServerConnection
  id: 9
  credential_type: basic
  params:
    database: trifacta
  type: microsoft_sqlserver
JSON results written to conn_create.out.
```

JSON Response:

```
{
    "conn_credential_location": "~/.trifacta/config_conn.json",
    "conn_credential_type": "basic",
    "conn_host": "example.com",
    "conn_id": 9,
    "conn_name": "aSQLServerConnection",
    "conn_params_location": "~/.trifacta/p.json",
    "conn_port": "1234",
    "conn_type": "microsoft_sqlserver",
    "host": "http://example.com:3005",
    "status": "success",
    "user_name": "<trifacta_user>"
}
```

Release 3.1.2:

Output:

```
Success: Connection aSQLServerConnection created
JSON results written to conn_create.out.
```

JSON Response:

```json
{
    "conn_credential_location": "~/.trifacta/config_conn.json",
    "conn_credential_type": "basic",
    "conn_host": "example.com",
    "conn_id": 9,
    "conn_name": "aSQLServerConnection",
    "conn_params_location": "~/.trifacta/p.json",
    "conn_port": "1234",
    "conn_type": "microsoft_sqlserver",
    "host": "http://example.com:3005",
    "results": {
        "createdAt": "2016-06-30T21:53:58.977Z",
        "createdBy": 3,
        "credential_type": "basic",
        "credentials": [
            {
                "username": "<trifacta_user>"
            }
        ],
        "deleted_at": null,
        "description": null,
        "host": "example.com",
        "id": 9,
        "is_global": false,
        "name": "aSQLServerConnection",
        "port": 1234,
        "type": "microsoft_sqlserver",
        "updatedAt": "2016-06-30T21:53:58.977Z",
        "updatedBy": 3
    },
    "status": "success",
    "user_name": "<trifacta_user>"
}
```

Edit connection

Example command (same for both releases):

```
./trifacta_cli.py edit_connection --user_name <trifacta_user> --password
<trifacta_password>
--conn_name aSQLServerConnection <--conn_type microsoft_sqlserver>
<--conn_description "This is my connection.">
<--conn_host mynewhost.com> <--conn_port 1234>
<--conn_credential_type basic> <--conn_credential_location ~/.trifacta
/config_conn.json>
<--cli_output_path ./conn_edit.out>
```

Release 3.1.1:

Output:

```
Updating connection aSQLServerConnection
Connection information for aSQLServerConnection
  description: This is my connection.
  host: mynewhost.com
  credentials: ["{u'username': u'<trifacta_user>'}"]
  port: 1234
  is_global: False
  name: aSQLServerConnection
  id: 9
  credential_type: basic
  params:
    database: trifacta
  type: microsoft_sqlserver
JSON results written to conn_edit.out.
```

JSON Response:

```
{
    "conn_description": "This is my connection.",
    "conn_id": 9,
    "conn_name": "aSQLServerConnection",
    "conn_params_location": "~/.trifacta/p.json",
    "host": "http://localhost:3005",
    "status": "success",
    "user_name": "<trifacta_user>"
}
```

Release 3.1.2:

Output:

```
Success: Updated connection aSQLServerConnection
JSON results written to conn_edit.out.
```

JSON Response:

```
{
    "conn_description": "This is my connection.",
    "conn_id": 9,
    "conn_name": "aSQLServerConnection",
    "conn_params_location": "~/.trifacta/p.json",
    "host": "http://nynewhost.com:3005",
    "results": {
        "createdAt": "2016-06-30T22:08:47.016Z",
        "createdBy": 3,
        "credential_type": "basic",
        "credentials": [
            {
                "username": "<trifacta_user>"
            }
        ],
        "deleted_at": null,
        "description": "This is my connection.",
        "host": "mynewhost.com",
        "id": 9,
        "is_global": false,
        "name": "aSQLServerConnection",
        "port": 1234,
        "type": "microsoft_sqlserver",
        "updatedAt": "2016-06-30T22:09:03.670Z",
        "updatedBy": 3
    },
    "status": "success",
    "user_name": "<trifacta_user>"
}
```

List connections

Example command (same for both releases):

```
./trifacta_cli.py list_connections --host dev.redshift.example.com
--user_name <trifacta_user> --password <trifacta_password>
--cli_output_path ./conn_list.out
```

Release 3.1.1:

Output:

```
   Listing connections
   Found 2 connections for params {'noLimit': 'true'}.
   Redshift:
     description: None
     host: dev.redshift.example.com
     credentials: ["{u'username': u'<trifacta_user>'}"]
     port: 5439
     is_global: True
     name: Redshift
     id: 2
     credential_type: custom
     params:
        extraLoadParams: BLANKSASNULL EMPTYASNULL TRIMBLANKS TRUNCATECOLUMNS
      defaultDatabase: dev
     type: amazon_redshift
   Hive:
     description: None
     host: dev.hive.example.com
     credentials: ["{u'username': u'<trifacta_user>'}"]
     port: 10000
     is_global: True
     name: Hive
     id: 1
     credential_type: conf
     params:
        jdbc: hive2
      connectStrOpts:
      defaultDatabase: default
     type: hadoop_hive
   JSON results written to conn_list.out.
```

JSON Response:

```
{
    "connections": [
        {
            "conn_createdAt": "2016-06-01T21:12:59.383Z",
            "conn_createdBy": 2,
            "conn_credential_type": "custom",
            "conn_credentials": [
                {
                    "username": "<trifacta_user>"
                }
            ],
            "conn_deleted_at": null,
            "conn_description": null,
            "conn_host": "dev.redshift.example.com",
            "conn_id": 2,
            "conn_is_global": true,
            "conn_name": "Redshift",
```

```
            "conn_params": {
                "extraLoadParams": "BLANKSASNULL EMPTYASNULL TRIMBLANKS
TRUNCATECOLUMNS",
                "defaultDatabase": "dev"
            },
            "conn_port": 5439,
            "conn_type": "amazon_redshift",
            "conn_updatedAt": "2016-06-01T21:33:38.672Z",
            "conn_updatedBy": 2
        },
        {
            "conn_createdAt": "2016-06-01T21:11:41.222Z",
            "conn_createdBy": 2,
            "conn_credential_type": "conf",
            "conn_credentials": [
                {
                    "username": "<trifacta_user>"
                }
            ],
            "conn_deleted_at": null,
            "conn_description": null,
            "conn_host": "dev.hive.example.com",
            "conn_id": 2,
            "conn_is_global": true,
            "conn_name": "Hive",
            "conn_params": {
                "jdbc": "hive2",
                "connectStrOpts": "",
                "defaultDatabase": "default"
            },
            "conn_port": 10000,
            "conn_type": "hadoop_hive",
            "conn_updatedAt": "2016-06-01T21:39:58.090Z",
            "conn_updatedBy": 2
        }
    ],
    "host": "http://localhost:3005",
    "status": "success",
    "user_name": "<trifacta_user>"
}
```

Release 3.1.2:

No changes.

Delete connection

Example command (same for both releases):

```
./trifacta_cli.py delete_connection --user_name <trifacta_user> --
password <trifacta_password>
--conn_name aSQLServerConnection --cli_output_path ./conn_delete.out
```

Release 3.1.1:

Output:

```
Deleting connection with id: 9
Delete successful.
JSON results written to conn_delete.out.
```

JSON Response:

```
{
    "conn_name": "aSQLServerConnection",
    "host": "http://localhost:3005",
    "status": "success",
    "user_name": "<trifacta_user>"
}
```

Release 3.1.2:

Output:

```
Success. Deleted connection with id: 9
JSON results written to conn_delete.out.
```

JSON Response:

Same as previous.

```
{
    "conn_name": "aSQLServerConnection",
    "host": "http://localhost:3005",
    "status": "success",
    "user_name": "<trifacta_user>"
}
```

Changes for Release 3.1

With the introduction of the Connection object model for publication, some parameters required for the Trifacta®
Command Line Interface (CLI) have been removed and replaced by other parameters.

> **NOTE:** This section identifies the changes that must be applied to your scripts for the Trifacta Command Line Interface if you are upgrading a Release 3.0.1 or earlier instance of the platform to Release 3.1 or later. Scripts that were functional on earlier versions of the platform do not function in Release 3.1 or later without these modifications.

See *CLI for Connections*.

See *CLI for Jobs*.

See *CLI for User Admin*.

CLI Tools Available as Executables

The command-line interface tools are now deployed as executables. Commands should continue to reference the Python scripts that were available in previous versions. These targets now serve as symlinks to the executables stored elsewhere in the Trifacta deployment.

Download Upgraded Transform Scripts

> **NOTE:** If you have upgraded from a previous version of the Trifacta platform where you were using the CLI, you must download again from the platform any transform scripts that you use in your CLI commands.

During the upgrade process, your transform scripts have been changed in their internal form, and the old versions are unlikely to work as expected with the command line interface.

You can download individual scripts through the Transformer page. See *Transformer Page*.

> **NOTE:** If you have a large number of transform scripts used by the CLI, please contact *Trifacta Support*.

Behavior Changes

Asynchronous publishing

Jobs are executed asynchronously now.

When you execute a publish operation, the platform immediately returns a response containing the job Id to monitor.

You can monitor progress of your publish operation using a `get_job_status` command.

Append and overwrite jobs for Hive

In Release 3.0.1 and earlier, you could publish to a new Hive table only using the `publish` action. After publication, no further updating was available through the Trifacta platform.

Beginning in Release 3.1, Hive supports the following two publication actions:

- `load_data` - Load data into the database table, to which a schema has already been applied. Use to append to existing table.
- `truncate_and_load` - Overwrite data in specified table.

The `publish` action is still supported, but its parameters have changed. See below.

Parameter Changes

Removed	Replaced With	Notes
`publication_target`	`connection_name`	In the new connection object model, publication targets are referred to using the internal name.
`disable_server_certificate _verification`	`disable_ssl_certification`	Shortened name

Localhost is assumed

If the `--host` parameter is not specified, the following parameter value is assumed:

```
--host localhost:3005
```

Release 3.0 and earlier documentation included this parameter and value, which are unnecessary.

> **NOTE:** If you are executing the CLI against a host other than `localhost` , the `host` parameter must be specified, including the port number. Port `3005` is the default value for the platform; to use that port, it must be included.

Unchanged commands

The parameters for the following commands have not changed between Release 3.0 and Release 3.1:

- `run_job`
- `get_job_status`

Example Commands

The following show examples of each type of command from Release 3.0. The equivalent for Release 3.1 is listed below it.

> **NOTE:** These commands apply to Release 3.1.0 only. They change again for Release 3.1.1 and later. For details, see *Changes for Release 3.1.1* below.
>
> For more information on your build number, select **User menu > About Trifacta** in the application.

create_schema action

Release 3.0 command (all one command):

```
./bin/trifacta_cli.py create_schema --user_name <trifacta_user> --
password <trifacta_password> --job_id 42
--database dev --table table_42 --cli_output_path ./create_info.out
```

Release 3.1 command (all one command):

```
./bin/trifacta_cli.py create_schema --user_name <trifacta_user> --
password <trifacta_password> --job_id 42
--database dev --table table_42 --connection_name aSQLServerConnection
--cli_output_path ./create_info.out
```

publish action

Release 3.0 command (all one command):

```
./bin/trifacta_cli.py publish --user_name <trifacta_user> --password
<trifacta_password> --job_id 42
--database dev --table table_job_42 --publish_format avro --
publication_target redshift
--cli_output_path ./publish_info.out
```

Release 3.1 command (all one command):

```
./bin/trifacta_cli.py publish --user_name <trifacta_user> --password
<trifacta_password> --job_id 42
--database dev --table table_job_42 --connection_name 1 --publish_format
avro
--cli_output_path ./publish_info.out
```

load_data action

Release 3.0 command (all one command):

```
./bin/trifacta_cli.py load_data --user_name <trifacta_user> --password
<trifacta_password> --job_id 42
--database dev --table table_42 --cli_output_path ./load_info.out --
publish_format avro
```

Release 3.1 command (all one command):

```
./bin/trifacta_cli.py load_data --user_name <trifacta_user> --password
<trifacta_password> --job_id 42
--database dev --table table_42 --connection_name aSQLServerConnection --
publish_format avro
--cli_output_path ./load_info.out
```

truncate_and_load action

Release 3.0 command:

- This action was not available in Release 3.0.

Release 3.1 command (all one command):

```
./bin/trifacta_cli.py truncate_and_load --user_name <trifacta_user> --
password <trifacta_password> --job_id 10
--database dev --table table_43 --connection_name aSQLServerConnection --
publish_format avro
--cli_output_path ./load_and_trunc_info.out
```

Connections actions

Beginning in Release 3.1, publishing targets are accessed through connection objects. Through the CLI, you can add, edit, list, or delete connections.

This capability did not exist prior to Release 3.1.

For more information on using the CLI to create connections, see *CLI for Connections*.

User Admin actions

There were no updates to CLI user administration actions in Release 3.1.

See *CLI for User Admin*.

Changes for Release 3.1.1

The following changes to the Command Line Interface have been applied to Release 3.1.1.

> **NOTE:** If you have upgraded from Release 3.0 or earlier, you should review first the changes that were made available in Release 3.1.0. See *Changes for Release 3.1*.

Improved error messaging

For Release 3.1.0 and earlier, some error messages were ambiguous or outright confusing. For example, multiple error states returned a message with _dict_ as part of the information.

For Release 3.1.1, error messages have been improved to provide more specific information about the issue and the parameters that were applied as part of the command.

Update to Hive params file

> **NOTE:** If you are upgrading from Release 3.1 or earlier and have been using the CLI to connect to Hive, this change is required.

In Release 3.0.1 and earlier, the Hive params file utilized a parameter called, `connectStringOptions` for passing arguments to Hive:

```
"connectStringOptions": ";transportMode=http;httpPath=cliservice"
```

In Release 3.1.1, the name of this parameter has been changed to `connectStrOpts`, which is more consistent with the internal storage of the parameter:

```
"connectStrOpts": ";transportMode=http;httpPath=cliservice"
```

Changes to CLI for Connections

For Release 3.1.1, a number of improvements have been applied to the CLI for connections.

> **The CLI for connections was introduced in Release 3.1.0.If you are upgrading from Release 3.1.0, these changes must be applied to each of your CLI scripts. For more information, see** *CLI for Connections*.

Updated standard output and JSON response for connection commands

For Release 3.1.1, the messages delivered back to standard output and in the JSON response from the Trifacta node have been made consistent with the parameter names entered at the command line.

> **Tip:** The names referenced in the JSON response should match the parameter names used in the command line interface. As of this release, you can use the JSON response as input for your next CLI command.

General structure of connection parameter names

Connection parameters are now prefaced with `--conn` instead of `--connection`. Some examples:

Release 3.1.0 example	Release 3.1.1 example
`--connection_type`	`--conn_type`
`--connection_host`	`--conn_host`

This general change applies to all connection parameters.

> **NOTE:** If you are upgrading from Release 3.1 and used the CLI for `publish` or `get_publications` commands, those scripts must be updated to use the new `conn_name` parameter. See *CLI for Jobs*.

Specific connection parameter name changes

Release 3.1.0 or earlier example	Release 3.1.1 example	Notes
`--connection_params_file`	`--conn_params_location`	Applies to upgrades from Release 3.1.0 only
`--connection_credential_file`	`--conn_credential_location`	Applies to upgrades from Release 3.1.0 only

Changes to connection example commands

Below are updates to example commands published in the Release 3.1.0 documentation.

Changes to Create Connection example

> **NOTE:** Applies to upgrades from Release 3.1.0 only.

Release 3.1.0 example command:

```
./trifacta_cli.py create_connection --user_name <trifacta_user> --
password <trifacta_password>
--connection_type microsoft_sqlserver --connection_name
aSQLServerConnection
--connection_description "This is my connection."
--connection_host example.com --connection_port 1234
--connection_credential_type basic
--connection_credential_file ~/.trifacta/config_conn.json
--connection_params_file ~/.trifacta/p.json
--cli_output_path ./connection_create.out
```

Release 3.1.1 example command:

```
./trifacta_cli.py create_connection --user_name <trifacta_user> --
password <trifacta_password>
--conn_type microsoft_sqlserver --conn_name aSQLServerConnection
--conn_description "This is my connection."
--conn_host example.com --conn_port 1234
--conn_credential_type basic
--conn_credential_location ~/.trifacta/config_conn.json
--conn_params_location ~/.trifacta/p.json
--cli_output_path ./conn_create.out
```

Changes to Edit Connection example

> **NOTE:** Applies to upgrades from Release 3.1.0 only.

Release 3.1.0 example command:

```
./trifacta_cli.py edit_connection --user_name <trifacta_user> --password
<trifacta_password>
--connection_name aSQLServerConnection <--connection_type
microsoft_sqlserver>
<--connection_description "This is my connection.">
<--connection_host mynewhost.com> <--connection_port 1234>
<--connection_credential_type basic> <--connection_credential_file ~/.
trifacta/config_conn.json>
<--cli_output_path ./connection_edit.out>
```

Release 3.1.1 example command:

```
./trifacta_cli.py edit_connection --user_name <trifacta_user> --password
<trifacta_password>
--conn_name aSQLServerConnection <--conn_type microsoft_sqlserver>
<--conn_description "This is my connection.">
<--conn_host mynewhost.com> <--conn_port 1234>
<--conn_credential_type basic> <--conn_credential_location ~/.trifacta
/config_conn.json>
<--cli_output_path ./conn_edit.out>
```

Changes to List Connections example

> **NOTE:** Applies to upgrades from Release 3.1.0 only.

There are no changes to the command to list all connections. Example:

```
./trifacta_cli.py list_connections --host dev.redshift.example.com
--user_name <trifacta_user> --password <trifacta_password>
--cli_output_path ./connection_list.out
```

If you have filtered any of your `list_connections` commands, please be sure to update your commands to the new parameter names. For example, if you are listing connections by name, you must change the parameter name of `--connection_name` to `-conn_name`.

Changes to Delete Connection example

> **NOTE:** Applies to upgrades from Release 3.1.0 only.

Release 3.1.0 example command:

```
./trifacta_cli.py delete_connection --user_name <trifacta_user> --
password <trifacta_password>
--connection_name aSQLServerConnection --cli_output_path .
/connection_delete.out
```

Release 3.1.1 example command:

```
./trifacta_cli.py delete_connection --user_name <trifacta_user> --
password <trifacta_password>
--conn_name aSQLServerConnection --cli_output_path ./conn_delete.out
```

Updates to connections documentation

In addition to the documentation changes for the above updates, the following items have been corrected in Release 3.1.1 documentation:

List connections by connection name

You can filter the list of connections by using the `connection_name` parameter. In Release 3.1.0 documentation, the `list_connections` command was not listed among the commands where `conn_name` applies.

Commands that can use connection identifiers

The `list_connections` and `delete_connections` commands can reference the connection to change by the internal connection identifier, which is defined when a connection is created.

In Release 3.1.0, the `conn_id` parameter was not documented.

Changes to the Admin Settings Page

Contents:

- *What Has Changed*
- *Accessing the Admin Settings Page*
    - *Minimum requirements*
    - *On running instance*
- *Limitations*
    - *Backend-only Parameters*
- *Mapping*
    - *Performance*
    - *Job Output*
    - *Hadoop Job Type*
    - *YARN*
    - *Hive*
    - *Pig*
    - *ZooKeeper*
    - *License*
    - *Enable/Disable Features*
    - *Users*
    - *Services*

> **In Release 4.1 and later, you should use the Admin Settings page for configuration changes whenever it is available. All parameters are accessible through this interface.**

What Has Changed

In Release 3.2.1 and earlier, the Admin Settings page contained a sampling of key parameters that administrators might want to change from within the web application.

In Release 4.0 and later, the Admin Settings page now contains a searchable list of parameters available through the web application. When parameter values are saved, they are written back to `trifacta-conf.json` on the Trifacta node .

This new version of the page allows administrators to make changes without having to manage access to the Trifacta node. Additionally, administrators can begin configuring the platform much earlier using the UI.

This section assists administrators who have upgraded from a previous version on how to migrate their configuration workflows to this new feature.

> **NOTE:** The Admin Settings page is the recommended method for changing configuration. Some parameters may not be available through this page. For more information, see
> *Platform Configuration Methods*.

In a running instance of the platform, select **User menu > Admin Settings**. For more information, see *Admin Settings Page*.

Accessing the Admin Settings Page

Minimum requirements

At a minimum, to use the Admin Settings page, please complete the following:

1. Install and initialize the databases. See *Install Databases*.
2. Install or upgrade the Trifacta software on the node. See *Install*.
3. See below for access.

On running instance

If you have installed or upgraded the software on the Trifacta node and verified that the software is connected to the database, you can begin using the Admin Settings page in the web application for further configuration.

Steps:

1. If you haven't done so already, start the platform. See *Start and Stop the Platform*.
2. Login to the application with an administrator account. See *Login*.
3. In the application menu, select **User Menu > Admin Settings**.

Limitations

> **Do not modify settings through the Admin Settings page and through `trifacta-conf.json` at the same time. Saving changes in one interface wipes out any unsaved changes in the other interface. Each requires a platform restart to apply the changes.**

The following are known limitations of this interface.

- Some parameters that are available in `trifacta-conf.json` are not available through the Admin Settings page.
- When you save changes in the Admin Settings page, the platform is automatically restarted.
- You can only edit parameters through this interface. You cannot add or delete parameters. You can set parameters to empty values.

Backend-only Parameters

The following parameters are not available through the Admin Settings page. These parameters must be changed through `trifacta-conf.json` on the Trifacta node. This list may not be complete.

```
hdfs.webhdfs.credential.password
smtp.*
```

Mapping

Below is the documentation from the Admin Settings page from Release 3.2.1.

For each section, a listing has been added to identify the relevant property names in the Release 4.0 version of the page.

> **Tip:** The values in each Release 4.0 table can be pasted into the search box in the Admin Settings page.

Performance

Browser Sample Size

Release 4.0	webapp.client.loadLimit

Limits the size in bytes of the data sample that is served back to the browser.

- For data that features wide columns or a high number of columns, you may need to increase the sample size.
- The hard limit of 2MB (2097152 bytes) prevents overwhelming the browser with data.

Max Result Download Size

Release 4.0	webapp.maxQueryResultsSize

Limits the maximum volume in bytes of any results downloaded from the application.

- Default is 10 GB.

Pig Progress Timeout

Release 4.0	batchserver.polling.progressTimeoutSeconds

Maximum time that a Pig job is allowed to run without making forward progress before it is killed.

> **NOTE:** As of Release 4.1, the Pig running environment is no longer available.

- Default is 3600 seconds (1 hour).

Max URL Encoded Upload

Release 4.0	webapp.bodyParser.urlEncoded.limit

This value defines the largest URL-encoded payload that can be sent from the client to the Trifacta platform.

> **NOTE:** Max JSON Upload and Max URL Encoded Upload should be modified together.

- Default value is 10MB.

Max JSON Upload

Release 4.0	webapp.bodyParser.json.limit

By default, the maximum size for a JSON body that can be sent from the client to the Trifacta platform is set to 10 MB. In some environments, individual operations may exceed this limit, which can cause memory failures or performance issues.

> **NOTE:** Max JSON Upload and Max URL Encoded Upload should be modified together.

In particular, this issue can appear when:

- you are performing multiple joins in your datasets
- you have a dataset with many columns
- your dataset contains wide column names

If needed, you can raise the maximum size of the JSON body sent to the platform.

> **NOTE:** Setting this value over 20 MB may cause requests to the platform to fail or significant performance degradation. You may need to experiment with values to find the right value.

Job Output

CSV Output Delimiter

Release 4.0	webapp.outputCsvDelimiter

When the application generates CSV output, the default field delimiter is the comma (,). As needed, you can change the delimiter.

Tableau CSV Limit

Release 4.0	webapp.maxResultSizeForTDEDownloadInMB

The maximum size in megabytes of the CSV file used for TDE Download. Default value is 40 MB.

Hadoop Job Type

Please select 'yarn'

> **NOTE:** As of Release 2.7, Map Reduce 1 has been deprecated. For more information, please see *End of Life and Deprecated Features*.

Select `yarn` to execute your jobs, and then complete the appropriate settings in the following sections.

YARN

Resource Manager Host

Release 4.0	yarn.resourcemanager.host

Host name of the node hosting the YARN resource manager.

Resource Manager Port

Release 4.0	yarn.resourcemanager.port

Port of the YARN resource manager to use.

Hive

Optionally, HIve can be used as a datastore for reading and writing datasets. See *Configure for Hive*.

Hive Server Host

Release 4.0	This value is managed through the connection that you create through the CLI to the enterprise Hive instance. For more information, see *CLI for Connections*.

Hostname of your Hive instance.

Hive Server Port

Release 4.0	This value is managed through the connection that you create through the CLI to the enterprise Hive instance. For more information, see *CLI for Connections* .

Port number through which to access your Hive instance. Default value is `10000`.

Pig

> **NOTE:** As of Release 4.1, the Pig running environment and Pig UDFs are no longer available in the platform.

UDF Jar

Release 4.0	batchserver.pig.udfJar

The Trifacta platform can be configured to use Pig scripts in a Hadoop environment. This setting must provide the relative path from the Trifacta deployment directory to the Pig user definition jarfile.

ZooKeeper

Release 4.0	ZooKeeper is no longer required by the Trifacta platform .

> **NOTE:** ZooKeeper is no longer required by the Trifacta platform. These settings will be removed in a future release.

Host

Host of the ZooKeeper node. If ZooKeeper was not available before Trifacta installation, this host is likely the Trifacta node.

Port

Port to use to access ZooKeeper.

License

License Location

Release 4.0	license.location

Specify the location of the Trifacta license key file. By default, this file is stored in the following location:

```
/license/license.json
```

This path must be specified relative to the top-level directory of the Trifacta deployment.

For more information, see *License Key*.

Enable/Disable Features

High Availability Hadoop Name Node

Release 4.0	feature.highAvailability.namenode

Toggles high availability for HDFS namenodes in the Hadoop cluster. Additional configuration is required. See *Enable Integration with Cluster High Availability*.

High Availability Hadoop Job Tracker

Release 4.0	feature.highAvailability.jobtracker

Toggles high availability for Jobtracker nodes in the Hadoop cluster. Additional configuration is required. See *Enable Integration with Cluster High Availability*.

Advanced automatic column split

Release 4.0	webapp.enableStructureDetection

For complex structured datasets, Trifacta Wrangler Enterprise can apply advanced algorithms to initially split the columns of your dataset.

Data Download

Release 4.0	webapp.enableDataDownload

By default, users can download generated results through the application. If needed, you can deselect this option to prevent data downloads. When it's disabled, results must be downloaded through the backend datastore.

> **NOTE:** Disabling this setting prevents all users from downloading data through the application, including admin users.

Users

Release 4.0	No changes to this configuration area.

You can manage aspects of user accounts through the Admin Settings page. See *Manage Users*.

Services

Release 4.0	No changes to this configuration area.

You can review overall status of the Trifacta platform.

Changes to the Object Model

Contents:

- *Release 6.0*
- *Release 5.1*
- *Release 5.0*
  - *Datasets with parameters*
- *Release 4.2*
  - *Wrangled datasets are removed*
  - *Recipes can be reused and chained*
  - *Introducing References*
  - *Introducing Outputs*
  - *Flow View Differences*
  - *Connections as a first-class object*
- *Release 4.1*
- *Release 4.0*
- *Release 3.2*
  - *Overview*
  - *Terminology Changes*
  - *Functional Changes*
  - *Changes to System Behavior due to Object Model Changes*

Beginning in Release 3.2, changes are being applied to the Trifacta® platform object model. These changes are intended to improve overall operationalization of the platform, enable better reuse of objects, and drive the platform toward a more flexible, workflow-based usage. These changes are to be applied over multiple releases.

These changes may have impacts on how you access features, although most features perform as expected from previous releases. In some cases:

- Features may behave differently.
- Features may be temporarily disabled in the current release, in favor of a new and improved implementation in a future release.
- Features may be removed altogether.

These changes are described in detail below.

For more information, see *Object Overview*.

Release 6.0

None.

Release 5.1

None.

Release 5.0

Datasets with parameters

Beginning in Release 5.0, imported datasets can be augmented with parameters, which enables operationalizing sampling and jobs based on date ranges, wildcards, or variables applied to the input path. For more information, see *Overview of Parameterization*.

Release 4.2

In Release 4.2, the object model has undergone the following revisions to improve flexibility and control over the objects you create in the platform.

Wrangled datasets are removed

In Release 3.2, the object model introduced the concepts of imported datasets, recipes, and wrangled datasets. These objects represented data that you imported, steps that were applied to that data, and data that was modified by those steps.

In Release 4.2, the wrangled dataset object has been removed in place of two objects listed below. All of the functionality associated with a wrangled dataset remains, including the following actions. Next to these actions are the new object with which the action is associated.

Wrangled Dataset action	Release 4.2 object
Run or schedule a job	Output object
Preview data	Recipe object
Reference to the dataset	Reference object

> **NOTE:** At the API level, the `wrangledDataset` endpoint continues to be in use. In a future release, separate endpoints will be available for recipes, outputs, and references. For more information, see *API Reference*.

These objects are described below.

Recipes can be reused and chained

Since recipes are no longer tied to a specific wrangled dataset, you can now reuse recipes in your flow. Create a copy with or without inputs and move it to a new flow if needed. Some cleanup may be required.

This flexibility allows you to create, for example, recipes that are applicable to all of your datasets for initial cleanup or other common wrangling tasks.

Additionally, recipes can be created from recipes, which allows you to create chains of recipes. This sequencing allows for more effective management of common steps within a flow.

Introducing References

Before Release 4.2, reference datasets existed and were represented in the user interface. However, these objects existed in the downstream flow that consumes the source. If you had adequate permissions to reference a dataset from outside of your flow, you could pull it in as a reference dataset for use.

In Release 4.2, a **reference** is a link between a recipe in your flow to other flows. This object allows you to expose your flow's recipe for use outside of the flow. So, from the source flow, you can control whether your recipe is available for use.

This object allows you to have finer-grained control over the availability of data in other flows. It is a dependent object of a recipe.

> **NOTE:** For multi-dataset operations such as union or join, you must now explicitly create a reference from the source flow and then union or join to that object. In previous releases, you could directly join or union to any object to which you had access.

Introducing Outputs

In Release 4.1, outputs became a configurable object that was part of the wrangled dataset. For each wrangled dataset, you could define one or more publishing actions, each with its own output types, locations, and other parameters. For scheduled executions, you defined a separate set of publishing actions. These publishing actions were attached to the wrangled dataset.

In Release 4.2, an **output** is a defined set of scheduled or ad-hoc publishing actions. With the removal of the wrangled dataset object, outputs are now top-level objects attached to recipes. Each output is a dependent object of a recipe.

Flow View Differences

Below, you can see the same flow as it appears in Release 4.1 and Release 4.2. In each Flow View:

- The same datasets have been imported.
- POS-r01 has been unioned to POS-r02 and POS-r03.
- POS-r01 has been joined to REF-PROD, and the column containing the duplicate join key in the result has been dropped.
- In addition to the default CSV publishing action (output), a scheduled one has been created in JSON format and scheduled for weekly execution.

Release 4.1 Flow View
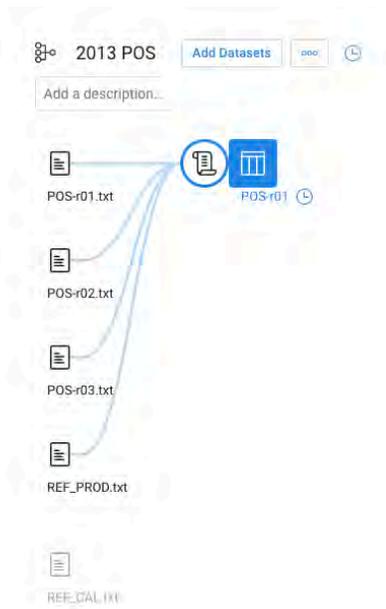
Release 4.2 Flow View

Flow View differences

- Wrangled dataset no longer exists.
- In Release 4.1, scheduling is managed off of the wrangled dataset. In Release 4.2, it is managed through the new output object.
    - Outputs are configured in a very similar manner, although in Release 4.2, the tab is labeled, "Destinations."
    - No changes to scheduling UI.
- Like the output object, the reference object is an externally visible link to a recipe in Flow View. This object just enables referencing the recipe object in other flows.
- See *Flow View Page*.

Other differences

- In application pages where you can select tabs to view object types, the available selections are typically: All, Imported Dataset, Recipe, and Reference.
- Wrangled datasets have been removed from the Dataset Details page, which means that the job cards for your dataset runs have been removed.
  - These cards are still available in the Jobs page when you click the drop-down next to the jjob entry.
  - The list of jobs for a recipe is now available through the output object in Flow View. Select the object and review the job details through the right panel.
- In Flow View and the Transformer page, context menu items have changed.

Connections as a first-class object

In Release 4.1.1 and earlier, connections appeared as objects to be created or explored in the Import Data page. Through the left navigation bar, you could create or edit connections to which you had permission to do so. Connections were also selections in the Run Job page.

- Only administrators could create public connections.
- End-users could create private connections.

In Release 4.2, the Connections Manager enables you to manage your personal connections and (if you're an administrator) global connections. Key features:

- Connections can be managed like other objects.
- Connections can be shared, much like flows.
  - When a flow with a connection is shared, its connection is automatically shared.
  - For more information, see *Overview of Sharing*.
- Release 4.2 introduces a much wider range of connectivity options.
  - Multiple Redshift connections can be created through this interface. In prior releases, you could only create a single Redshift connection, and it had to be created through the command line interface (CLI).

> **NOTE:** Beginning in Release 4.2, all connections are initially created as **private connections**, accessible only to the user who created. Connections that are available to all users of the platform are called, public connections. You can make connections public through the Connections page.

For more information, see *Connections Page*.

Release 4.1

None.

Release 4.0

None.

Release 3.2

Overview

In Release 3.2, the object model has been moved from a dataset-oriented structure to a flow-based structure. Previously, datasets created in the application represented the central data objects. In the new flow-based model, all datasets that have been touched in the application are contained in a new object, called a flow. A flow is essentially a replacement of the project object with a different set of behaviors, including automatic change propagation between datasets. In the future, it will support even greater flexibility and connectivity.

Similarly, scripts created in the Transformer page are now called recipes, which will become much more flexible and reusable objects in the future.

In prior releases, datasources were references to source data that existed outside of the application and were controlled by Trifacta administrators. Beginning in Release 3.2, these objects, now called **imported datasets**, are independent objects and are associated with the dataset that uses them. They can be managed by the user who imports the data into the application.For an overview diagram, see *Object Overview*.

Terminology Changes

Old Term	New Term	Notes
project	flow	A **flow** is a more generalized container for datasets, which will enable greater reuse of assets. See *Flows Page*.
script	recipe	A **recipe** contains all of the transform steps of a script, as well as interfaces for their reuse in other scripts and datasets. See *Recipe Panel*.
datasource	imported dataset	An **imported dataset** is one or more files imported from outside of the platform. Functionally identical to the datasource in previous releases. Imported datasets can be associated with your flow.
dataset	Wrangled dataset	To distinguish from an imported dataset, a **Wrangled dataset** refers to any dataset that has been opened and edited in the Transformer page. A Wrangled dataset is a separate object in your flow.
execution engine	running environment	This is a simple terminology change. When you configure your jobs, you select the appropriate running environment, where your job is executed.See *Run Job Page*.

For more information on these changes, see *Object Overview*.

Functional Changes

Old Feature	New Feature	Description
Projects page	Flows page	Projects have been replaced by flows, which will offer much broader functionality and connectivity over the course of several releases. A flow is a storage container for imported and Wrangled datasets. See *Flows Page*.

Datasets page	Datasets page	The Datasets page is used to import data from an outside source. In Release 3.2 and later, you interact with both imported datasets and Wrangled datasets through the Datasets page. The workflow has changed a little bit. See *Library Page*.
Datasources page	REMOVED	Imported datasets (datasources) are now managed through the Datasets page. This page is no longer available in the application.
N/A	Dependencies Browser	Explore the dependencies in your datasets through the Transformer page. Identify dependency issues in the target dataset and then quickly navigate to the source issue to fix it. See *Recipe Navigator*.

Changes to System Behavior due to Object Model Changes

Automatic change propagation: Changes in one dataset automatically propagate to dependent datasets.

Imported and Wrangled datasets can be integrated into other datasets at any time. The changes to the object model support the propagation of changes in one dataset to be automatically applied in any datasets that consume the source dataset. This applies to the following:

- Joins
- Lookups
- Unions

> **NOTE:** This propagation does not apply to:
>
> 1. Datasets that are created from the generated output of the dataset. Since the new dataset is the product of an executed job, it no longer has any connection to the changes in the source dataset. If you wish to propagate those changes, however, you can re-run the job and write out a new dataset. See *Publishing Dialog* .
>
> 2. Copies of the dataset. Dataset copies are independent objects.

Implications:

In Release 3.1.2 and earlier, multi-dataset operations, such as union, join, and lookup, were executed on a snapshot of the other dataset. For example, if dataset A performed a lookup into dataset B, the application internally performed a snapshot of dataset B and used the snapshot for completing the lookup. This snapshot is maintained separately.

When the platform is upgraded to Release 3.2 and later, this snapshotting behavior is preserved. Instead of maintaining the internal snapshot, the snapshot is migrated into a wrangled dataset of the same name.

> **NOTE:** If your upgraded datasets included multi-dataset operations, you will see additional copies of the dataset that is used in the join or union. This dataset is saved such that the pre-migration snapshot is preserved. This method maintains the pre-upgrade state of the dataset and disables change propagation on the affected dataset.
>
> If desired, you can edit this dataset or switch to the true source dataset to enable automated change propagation.

Additional impacts of automated change propagation of specific multi-dataset operations:

- **Joins:** In prior releases, joins were executed on a snapshot of data. With automated change propagation, snapshotting is no longer necessary. The target dataset is automatically updated with any changes to the joined-in dataset.

> **NOTE:** Automated change propagation can cause breakages in downstream datasets. For example, if you make changes to a dataset that is used in a join, those changes can break steps in the dataset into which it is joined. The Recipe panel can be used to identify these issues, which you can navigate to fix through the new Dependencies browser in the Transformer page. See *Recipe Navigator*.

  - See *Join Panel*.
- **Lookups:** Similar to joins, changes in lookup data are automatically propagated. See *Lookup Wizard*.
- **Unions:** In prior releases, when a dataset that was part of a `union` transform was changed, an alert appeared in the Recipe panel of the target dataset to indicate that there was a change. Beginning in Release 3.2:

  - The data is automatically updated in the target dataset.
  - If the changes cause breakages, you can see the effects and the source dataset in the Recipe panel for the target dataset.
  - You can trace back these issues through the Dependencies browser. See *Recipe Navigator*.

Sharing disabled: Datasets cannot be shared between users for now.

This functionality will be replaced by more robust sharing capabilities in a future release.

> **Tip:** You can make datasets available to other users by generating results from the source and then pointing other users to the generated results to use import into their flow. However, this is not sharing, as changes to the source dataset are not propagated to any datasets generated from the output.

Random samples generated on previous releases are no longer available in upgraded systems.

Due to changes in the object model and other factors, random samples that you created on your datasets in previous releases are no longer available in Release 3.2.

When you first open a dataset in the Transformer page in Release 3.2, a new random sample is automatically generated for you. Also, given the larger sample size in Release 3.2, your entire dataset may be displayed in the Transformer page.

Undo/redo of dataset swap has been removed.

Changes to the object model mean that you cannot use undo/redo controls in the Transformer page to change the dataset.

> **Tip:** You can still select the previous dataset.

Improvements to the Type System

Contents:

This section provides information on improvements to the Trifacta® type system.

> **If you have upgraded from a Trifacta Release 3.0 or earlier to Release 3.1 or later, you should review this page, as some type-related behaviors have changed in the platform.**

General Improvements in Typecasting

Mismatched data types

Where there are mismatches between inputs and the expected input data type, the following values are generated for the mismatches:

Source data type	Output if mismatched
Primitive data types: - Integer - Decimal - Boolean - Arrays - Maps	null value, if mismatched
Datetime	null value, if mismatched
Other non-primitive data types, including: - SSN - Phone Number - Email Address - Credit Card - Gender - IP Address - URL - HTTP Code - Zip Code	Converted to string values, if mismatched
String	Anything can be a String value.

State values and custom data types are converted to string values, if they are mismatched.

Three-value logic for null values

The Trifacta Photon running environment has been augmented to use three-value logic for null values.

When values are compared, the result can be `true` or `false` in most cases.

If a null value was compared to a null value in the Trifacta Photon running environment:

- In Release 3.0 and earlier, this evaluated to `true`.
- In Release 3.1 and later, this evaluates to an unknown (null) value.

This change aligns the behavior of the running environment with that of SQL and Hadoop Pig.

Improved handling of null values

Assume that the column `nuller` contains null values and that you have the following transform:

```
derive value:(nuller >= 0)
```

Prior to Release 3.1, the above transform generated a column of `true` values.

In Release 3.1 and later, the transform generates a column of null values.

More consistent evaluation of null values in ternaries

In the following example, `a_null_expression` always evaluates to a null value.

```
derive value: (a_null_expression ? 'a' : 'b')
```

In Release 3.0, this expression generated `b` for all inputs on the Trifacta Photon running environment and a null value on Hadoop Pig.

In Release 3.1 and later, this expression generates a null value for all inputs on both running environments.

> **Tip:** Beginning in Release 3.1, you can use the `if` function instead of ternary expressions. Ternaries may be deprecated at some point in the future. For more information, see *IF Function*.

For example, you have the following dataset:

MyStringCol

| This works. |
| You can't break this. |
| Not broken yet. |

You test each row for the presence of the string `can't`:

```
derive value: if(find(MyStringCol, 'can\'t',true,0) > -1, true, false) as:'MyFindResults'
```

The above transform results in the following:

MyStringCol	MyFindResults
This works.	
You can't break this.	true
Not broken yet.	

In this case, the value of `false` is not written to the other columns, since the `find` function returns a null value. This null value, in turn, nullifies the entire expression, resulting in a null value written in the new column.

You can use the following to locate the null values:

```
derive value:isnull(MyFindResults) as:'nullInMyFindResults'
```

Datetime changes

Raw date and time values must be properly formatted

> **NOTE:** Upgraded recipes continue to function properly. However, if you edit the recipe step in an upgraded system, you are forced to fix the formatting issue before saving the change.

Before this release, you could create a transform like the following:

```
derive value:date(2016,2,15)
```

This transform generated a column of map values, like the following:

```
{"year":"2016","month":"2","date":"15"}
```

Beginning this release, the above command is invalid, as the date values must be properly formatted prior to display. The following works:

```
derive value:dateformat(date(2016,2,15),'yyyy-MM-dd')
```

This transform generates a column of Datetime values in the following format:

```
2016-02-15
```

Time:

Before this release:

```
derive value:time(11,34,58)
```

Prior release output:

```
{"hours":"11","minutes":"34","seconds":"58"}
```

This release:

```
derive value:dateformat(time(11,34,58), 'HH-mm-ss')
```

This release's output:

```
11-34-58
```

- See *DATEFORMAT Function*.
- See *UNIXTIMEFORMAT Function*.

Date formatting functions supports 12-hour time only if AM/PM indicator is included

Beginning in this release, the `unixtimeformat` and `dateformat` functions requires an AM/PM indicator (`a`) if the date formatting string uses a 12-hour time indicator (`h` or `hh`).

Valid for earlier releases:

```
derive value: unixtimeformat(myDate, 'yyyy-MM-dd hh:mm:ss') as:'myUnixDate'
```

Valid for this release and later:

```
derive value: unixtimeformat(myDate, 'yyyy-MM-dd hh:mm:ss a') as:'myUnixDate'
```

These references in recipes fail to validate in this release or later and must be fixed.

- See *DATEFORMAT Function*.
- See *Unixtimeformat Function*.

Un-inferrable formats from dateformat and unixtimeformat functions are written as strings

If a formatting string is not a datetime format recognized by the Trifacta platform, the output is generated as a string value.

This change was made to provide clarity to some ambiguous conditions.

Colon as a delimiter for date values is no longer supported

Beginning in this release, the colon (`:`) is no longer supported as a delimiter for date values. It is still supported for time values.

myDateValue	Recognized?
02:03:2016	No
02:03:16	Recognized as a time value

When data such as the above is imported, it may not be initially recognized by the Trifacta application as Datetime type.

To fix, you might apply the following transform:

```
replace col:myDateValue with:'-' on:`-` global:true
```

The new column values are more likely to be inferred as Datetime values. If not, you can choose the appropriate Datetime format from the data type drop-down for the column. See *Data Grid Panel*.

Changes to the User-Defined Functions

> **This documentation applies to Release 4.0. As of Release 4.1, the Pig running environment is no longer available. You can follow these steps on a pre-Release 4.1 instance to convert your Python UDFs to Java UDFs. This content will be removed in a future release.**

Deprecation of Python UDFs

> **As Release 4.1, Python UDFs are no longer available in the platform. You must transition or author UDFs in Java. Details are below.**

In Release 4.0 introduces Spark as the default running environment for Hadoop job execution. This environment does not support the execution of Python UDFs.

> **NOTE:** Environments that have been upgraded to Release 4.0 are automatically switched to using Spark as the default running environment. To complete these steps, you must re-enable Pig as the default running environment, which is described below.

This section describes the general process for converting existing Python UDFs to Java UDFs in a production environment.

> **NOTE:** Development of user-defined functions is a developer-level skill.

Steps:

1. After you have upgraded to Release 4.0, you must re-enable Pig for job execution. See *Running Environment Options*.
2. For each of your existing Python UDFs, author a new one in Java:
    1. This new UDF must have the same name and schema.
    2. Deploy each new Java UDF to the Trifacta node.
    3. For more information on authoring and deployment, see *Java UDFs*.
3. When all of your existing Python UDFs have been converted to Java and installed on the Trifacta node, enable the Spark running environment on the node. See *Running Environment Options*.
4. At this point:
    1. When you select **Run on Trifacta Server** , the Trifacta Photon running environment uses the Python UDFs.
        1. In later versions, this selection is **Photon**.
    2. When you select **Run on Hadoop**, the Spark running environment uses the Java UDFs.

1. In later versions, this selection is **Spark**.
5. Verify job results of executions on each running environment for each of your Python UDFs.
6. Remove the Python UDFs from the Trifacta node.

For Release 4.1 upgrade:

- Python UDF service will be removed.
- All Python UDFs will be deleted from the platform.

TODAY and NOW functions are part of application

In previous releases, the Python UDF documentation included examples on how to create the TODAY() and NOW() functions. These functions are now available through the application by default.

- See *TODAY Function*.
- See *NOW Function*.

Release Notes 6.0

Contents:

- *Release 6.0.2*
    - *What's New*
    - *Changes to System Behavior*
    - *Key Bug Fixes*
    - *New Known Issues*
- *Release 6.0.1*
    - *What's New*
    - *Changes to System Behavior*
    - *Key Bug Fixes*
    - *New Known Issues*
- *Release 6.0*
    - *What's New*
    - *Changes to System Behavior*
    - *Key Bug Fixes*
    - *New Known Issues*

Release 6.0.2

This release addresses several bug fixes.

What's New

- Support for Cloudera 6.2. For more information, see *System Requirements*.

Changes to System Behavior

> **NOTE:** As of Release 6.0, all new and existing customers must license, download, and install the latest version of the Tableau SDK onto the Trifacta node. For more information, see *Create Tableau Server Connections*.

Upload:

- In previous releases, files that were uploaded to the Trifacta platform that had an unsupported filename extension received a warning before upload.
- Beginning in this release, files with unsupported extensions are blocked from upload.
- You can change the list of supported file extensions. For more information, see *Miscellaneous Configuration*.

Documentation:

- In Release 6.0.x documentation, documentation for the API JobGroups Get Status v4 endpoint was mistakenly published. This endpoint does not exist. For more information on the v4 equivalent, see *Changes to the APIs*.

Key Bug Fixes

Ticket	Description
TD-40471	SAM auth: Logout functionality not working
TD-39318	Spark job fails with parameterized datasets sourced from Parquet files
TD-39213	Publishing to Hive table fails

New Known Issues

None.

Release 6.0.1

This release features support for several new Hadoop distributions and numerous bug fixes.

What's New

Connectivity:

- Support for integration with CDH 5.16.
- Support for integration with CDH 6.1. Version-specific configuration is required.

> **NOTE:** If you have upgraded to Cloudera 6.0.0 or later and are using EC2 role-based authentication to access AWS resources, you must change two platform configuration properties. For more information, see *Configure for EC2 Role-Based Authentication*.

  See *Supported Deployment Scenarios for Cloudera*.
- Support for integration with HDP 3.1. Version-specific configuration is required. See *Supported Deployment Scenarios for Hortonworks*.
  - Support for Hive 3.0 on HDP 3.0 or HDP 3.1. Version-specific configuration is required. See *Configure for Hive*.
- Support for Spark 2.4.0.

> **NOTE:** There are some restrictions around which running environment distributions support and do not support Spark 2.4.0.

  For more information, see *Configure for Spark*.
- Support for integration with high availability for Hive.

> **NOTE:** High availability for Hive is supported on HDP 2.6 and HDP 3.0 with Hive 2.x enabled. Other configurations are not currently supported.

For more information, see *Create Hive Connections*.

Publishing:

- Support for automatic publishing of job metadata to Cloudera Navigator.

> **NOTE:** For this release, Cloudera 5.16 only is supported.

For more information, see *Configure Publishing to Cloudera Navigator*.

Changes to System Behavior

Key Bug Fixes

Ticket	Description
TD-39779	MySQL JARs must be downloaded by user. > **NOTE:** If you are installing the databases in MySQL, you must download a set of JARs and install them on the Trifacta node. For more information, see *Install Databases for MySQL*.
TD-39694	Tricheck returns status code 200, but there is no response. It does not work through Admin Settings page.
TD-39455	HDI 3.6 is not compatible with Guava 26.
TD-39086	Hive ingest job fails on Microsoft Azure.

New Known Issues

Ticket	Description
TD-40299	Cloudera Navigator integration cannot locate the database name for JDBC sources on Hive.
TD-40348	When loading a recipe in imported flow that references an imported Excel dataset, Transformer page displays Input validation failed: (Cannot read property 'filter' of undefined) error, and the screen is blank. > **Workaround:** In Flow View, select an output object, and run a job. Then, load the recipe in the Transformer page and generate a new sample. For more information, see *Import Flow*.

| TD-39969 | On import, some Parquet files cannot be previewed and result in a blank screen in the Transformer page. |
| | **Workaround:** Parquet format supports row groups, which define the size of data chunks that can be ingested. If row group size is greater than 10 MB in a Parquet source, preview and initial sampling does not work. To workaround this issue, import the dataset and create a recipe for it. In the Transformer page, generate a new sample for it. For more information, see *Parquet Data Type Conversions*. |

Release 6.0

This release of Trifacta® Wrangler Enterprise introduces key features around column management, including multi-select and copy and paste of columns and column values. A new Job Details page captures more detailed information about job execution and enables more detailed monitoring of in-progress jobs. Some relational connections now support publishing to connected databases. This is our largest release yet. Enjoy!

NOTE: This release also announces the deprecation of several features, versions, and supported extensions. Please be sure to review Changes to System Behavior below.

What's New

NOTE: Beginning in this release, the Wrangler Enterprise desktop application requires a 64-bit version of Microsoft Windows. For more information, see *Install Desktop Application*.

Wrangling:

- In data grid, you can select multiple columns before receiving suggestions and performing transformations on them. For more information, see *Data Grid Panel*.
    - New Selection Details panel enables selection of values and groups of values within a selected column. See *Selection Details Panel*.
- Copy and paste columns and column values through the column menus. see *Copy and Paste Columns*.
- Support for importing files in Parquet format. See *Supported File Formats*.
- Specify ranges of key values in your joins. See *Configure Range Join*.

Jobs:

- Review details and monitor the status of in-progress jobs through the new Job Details page. See *Job Details Page*.
- Filter list of jobs by source of job execution or by date range. See *Jobs Page*.

Connectivity:

- Publishing (writeback) is now supported for relational connections.
    - This feature is enabled by default

        NOTE: After a connection has been enabled for publishing, you cannot disable publishing for that connection. Before you enable, please verify that all user accounts accessing databases of these types have appropriate permissions.

See *Enable Relational Connections*.
- The following connection types are natively supported for publishing to relational systems.
  - *Oracle Data Type Conversions*
  - *Postgres Data Type Conversions*
  - *SQL Server Data Type Conversions*
  - *Teradata Data Type Conversions*
- Import folders of Microsoft Excel workbooks. See *Import Excel Data*.
- Support for integration with CDH 6.0. Version-specific configuration is required.See *Supported Deployment Scenarios for Cloudera*.
- Support for integration with HDP 3.0. Version-specific configuration is required. See *Supported Deployment Scenarios for Hortonworks*.
  - Support for Hive 3.0 on HDP 3.0 only. Version-specific configuration is required. See *Configure for Hive*.

Language:

- Track file-based lineage using `$filepath` and `$sourcerownumber` references. See *Source Metadata References*.
- In addition to directly imported files, the `$sourcerownumber` reference now works for converted files (such as Microsoft Excel workbooks) and for datasets with parameters. See *Source Metadata References*.

Workspace:

- Organize your flows into folders. See *Flows Page*.

Publishing:

- Users can be permitted to append to Hive tables when they do not have CREATE or DROP permissions on the schema.

> **NOTE:** This feature must be enabled. See *Configure for Hive*.

Administration:

- New Workspace Admin page centralizes many of the most common admin settings. See Changes to System Behavior below.
- Download system logs through the Trifacta application. See *Admin Settings Page*.

Supportability:

- High availability for the Trifacta node is now generally available. See *Install for High Availability*.

Authentication:

- Integrate SSO authentication with enterprise LDAP-AD using platform-native LDAP support.

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

> **NOTE:** In previous releases, LDAP-AD SSO utilizes an Apache reverse proxy. While this method is still supported, it is likely to be deprecated in a future release. Please migrate to using the above SSO method. See *Configure SSO for AD-LDAP*.

- Support for SAML SSO authentication. See *Configure SSO for SAML*.

Changes to System Behavior

> **NOTE:** The Trifacta node requires NodeJS 10.13.0. See *System Requirements*.

Configuration:

To simplify configuration of the most common feature enablement settings, some settings have been migrated to the new Workspace Admin page. For more information, see *Workspace Admin Page*.

> **NOTE:** Over subsequent releases, more settings will be migrated to the Workspace Admin page from the Admin Settings page and from `trifacta-conf.json`. For more information, see *Changes to Configuration*.

See *Platform Configuration Methods*.

See *Admin Settings Page*.**Java 7:**

> **NOTE:** In the next release of Trifacta Wrangler Enterprise, support for Java 7 will be end of life. The product will no longer be able to use Java 7 at all. Please upgrade to Java 8 on the Trifacta node and your Hadoop cluster.

Key Bug Fixes

Ticket	Description
TD-36332	Data grid can display wrong results if a sample is collected and dataset is unioned.
TD-36192	Canceling a step in recipe panel can result in column menus disappearing in the data grid.
TD-35916	Cannot logout via SSO
TD-35899	A deployment user can see all deployments in the instance.
TD-35780	Upgrade: Duplicate metadata in separate publications causes DB migration failure.
TD-35644	Extractpatterns with "HTTP Query strings" option doesn't work.
TD-35504	Cancel job throws 405 status code error. Clicking Yes repeatedly pops up Cancel Job dialog.
TD-35486	Spark jobs fail on LCM function that uses negative numbers as inputs.
TD-35483	Differences in how WEEKNUM function is calculated in the Trifacta Photon and Spark running environments, due to the underlying frameworks on which the environments are created. **NOTE:** Trifacta Photon and Spark jobs now behave consistently. Week 1 of the year is the week that contains January 1.

TD-35481	Upgrade Script is malformed due to SplitRows not having a Load parent transform.
TD-35177	Login screen pops up repeatedly when access permission is denied for a connection.
TD-27933	For multi-file imports lacking a newline in the final record of a file, this final record may be merged with the first one in the next file and then dropped in the Trifacta Photon running environment.

New Known Issues

Ticket	Description
TD-39513	Import of folder of Excel files as parameterized dataset only imports the first file, and sampling may fail. **Workaround:** Import as separate datasets and union together.
TD-39455	HDI 3.6 is not compatible with Guava 26.
TD-39092	`$filepath` and `$sourcerownumber` references are not supported for Parquet file inputs. **Workaround:** Upload your Parquet files. Create an empty recipe and run a job to generate an output in a different file format, such as CSV or JSON. Use that output as a new dataset. See *Build Sequence of Datasets*. For more information on these references, see *Source Metadata References*.
TD-39086	Hive ingest job fails on Microsoft Azure.
TD-39053	Cannot read datasets from Parquet files generated by Spark containing nested values. **Workaround:** In the source for the job, change the data types of the affected columns to String and re-run the job on Spark.
TD-39052	Signout using reverse proxy method of SSO is not working after upgrade.
TD-38869	Upload of Parquet files does not support nested values, which appear as null values in the Transformer page. **Workaround:** Unnest the values before importing into the platform.
TD-37683	Send a copy does not create independent sets of recipes and datasets in new flow. If imported datasets are removed in the source flow, they disappear from the sent version. **Workaround:** Create new versions of the imported datasets in the sent flow.

TD-36145	Spark running environment recognizes numeric values preceded by + as Integer or Decimal data type. Trifacta Photon running environment does not and types these values as strings.
TD-35867	v3 publishing API fails when publishing to alternate S3 buckets
	Workaround: You can use the corresponding v4 API to perform these publication tasks. For more information on a workflow, see *API Workflow - Manage Outputs*.

Release Notes 5.1

Contents:

- *What's New*
- *Changes to System Behavior*
- *Key Bug Fixes*
- *New Known Issues*

Welcome to Release 5.1 of Trifacta® Wrangler Enterprise! This release includes a significant expansion in database support and connectivity with more running environment versions, such as Azure Databricks. High availability is now available on the Trifacta platform node itself.

Within the Transformer page, you should see a number of enhancements, including improved toolbars and column menus. Samples can be named.

Regarding operationalization of the platform, datasets with parameters can now accept Trifacta patterns for parameter specification, which simplifies the process of creating complex matching patterns. Additionally, you can swap out a static imported dataset for a dataset with parameters, which enables development on a simpler dataset before expanding to a more complex set of sources. Variable overrides can now be applied to scheduled job executions, and you can specify multiple variable overrides in Flow View.

The underlying language has been improved with a number of transformations and functions, including a set of transformations designed around preparing data for machine processing. Details are below.

> **Tip:** For a general overview of the product and its capabilities, see *Product Overview*.

What's New

Install:

- Support for PostgreSQL 9.6 for Trifacta databases.

  > **NOTE:** PostgreSQL 9.3 is no longer supported. PostgreSQL 9.3 is scheduled for end of life (EOL) in September 2018. For more information on upgrading, see *Upgrade Databases for PostgreSQL*.

- Partial support for MySQL 5.7 for hosting the Trifacta databases.

> **NOTE:** MySQL 5.7 is not supported for installation on Amazon RDS. See *System Requirements*.

- Support for high availability on the Trifacta node. See *Configure for High Availability*.

> **NOTE:** High availability on the platform is in Beta release.

- Support for CDH 5.15.

> **NOTE:** Support for CDH 5.12 has been deprecated. See *End of Life and Deprecated Features*.

- Support for Spark 2.3.0 on the Hadoop cluster. See *System Requirements*.
- Support for integration with EMR 5.13, EMR 5.14, and EMR 5.15. See *Configure for EMR*.

> **NOTE:** EMR 5.13 - 5.15 require Spark 2.3.0. See *Configure for Spark*.

- Support for integration with Azure Databricks. See *Configure for Azure Databricks*.
- Support for WebAssembly, Google Chrome's standards-compliant native client.

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

> **NOTE:** In a future release, use of PNaCl native client is likely to be deprecated.
>
> Use of WebAssembly requires Google Chrome 68+. No additional installation is required. In this release, this feature must be enabled. For more information, see *Miscellaneous Configuration*.

- The Trifacta® platform defaults to using Spark 2.3.0 for Hadoop job execution. See *Configure for Spark*.

Connectivity:

- Enhanced import process for Excel files, including support for import from backend file systems. See *Import Excel Data*.
- Support for DB2 connections. See *Connection Types*.
- Support for HiveServer2 Interactive (Hive 2.x) on HDP 2.6. See *Configure for Hive*.
- Support for Kerberos-delegated relational connections. See *Enable SSO for Relational Connections*.

> **NOTE:** In this release, only SQL Server connections can use SSO. See *Create SQL Server Connections*.

- Performance caching for JDBC ingestion. See *Configure JDBC Ingestion*.
- Enable access to multiple WASB datastores. See *Enable WASB Access*.

Import:

- Support for use of Trifacta patterns in creating datasets with parameters. See *Create Dataset with Parameters*.
- Swap a static imported dataset with a dataset with parameters in Flow View. See *Flow View Page*.

Flow View:

- Specify overrides for multiple variables through Flow View. See *Flow View Page*.
- Variable overrides can also be applied to scheduled job executions. See *Add Schedule Dialog*.

Transformer Page:

- Join tool is now integrated into the context panel in the Transformer page. See *Join Panel*.
    - Improved join inference key model. See *Join Panel*.
- Patterns are available for review and selection, prompting suggestions, in the context panel.
- Updated toolbar. See *Transformer Toolbar*.
- Enhanced options in the column menu. See *Column Menus*.
- Support for a broader range of characters in column names. See *Rename Columns*.

Sampling:

- Samples can be named. See *Samples Panel*.
- Variable overrides can now be applied to samples taken from your datasets with parameters. See *Samples Panel*.

Jobs:

- Filter list of jobs by date. See *Jobs Page*.

Language:

- Rename columns using values across multiple rows. See *Rename Columns*.

Transformation Name	Description
Bin column	Place numeric values into bins of equal or custom size for the range of values.
Scale column	Scale a column's values to a fixed range or to zero mean, unit variance distributions.
One-hot encoding	Encode values from one column into separate columns containing 0 or 1, depending on the absence or presence of the value in the corresponding row.
Group By	Generate new columns or replacement tables from aggregate functions applied to grouped values from one or more columns.

Publishing:

- Export dependencies of a job as a flow. See *Flow View Page*.
- Add quotes as CSV file publishing options. See *Run Job Page*.
- Specify CSV field delimiters for publishing. See *Miscellaneous Configuration*.
- Support for publishing Datetime values to Redshift as timestamps. See *Redshift Data Type Conversions*.

Execution:

- UDFs are now supported for execution on HDInsight clusters. See *Java UDFs*.

Admin:

- Enable deletion of jobs. See *Miscellaneous Configuration*.
- Upload an updated license file through the application. See *Admin Settings Page*.

Changes to System Behavior

Diagnostic Server removed from product

The Diagnostic Server and its application page have been removed from the product. This feature has been superseded by Tricheck, which is available to administrators through the application. For more information, see *Admin Settings Page*.

Wrangle now supports nested expressions

The Wrangle now supports nested expressions within expressions. For more information, see *Changes to the Language*.

Language changes

- The `RAND` function without parameters now generates true random numbers.
- When the source information is not available, the `SOURCEROWNUMBER` function can still be used. It returns null values in all cases.
- New functions.
- See *Changes to the Language*.

Key Bug Fixes

Ticket	Description
TD-36332	Data grid can display wrong results if a sample is collected and dataset is unioned.
TD-36192	Canceling a step in recipe panel can result in column menus disappearing in the data grid.
TD-36011	User can import modified exports or exports from a different version, which do not work.
TD-35916	Cannot logout via SSO
TD-35899	A deployment user can see all deployments in the instance.
TD-35780	Upgrade: Duplicate metadata in separate publications causes DB migration failure.
TD-35746	`/v4/importedDatasets` GET method is failing.
TD-35644	Extractpatterns with "HTTP Query strings" option doesn't work
TD-35504	Cancel job throws 405 status code error. Clicking Yes repeatedly pops up Cancel Job dialog.
TD-35481	After upgrade, recipe is malformed at splitrows step.
TD-35177	Login screen pops up repeatedly when access permission is denied for a connection.
TD-34822	Case-sensitive variations in date range values are not matched when creating a dataset with parameters. **NOTE:** Date range parameters are now case-insensitive.

TD-33428	Job execution on recipe with high limit in split transformation due to Java Null Pointer Error during profiling.
	NOTE: Avoid creating datasets that are wider than 2500 columns. Performance can degrade significantly on very wide datasets.
TD-31327	Unable to save dataset sourced from multi-line custom SQL on dataset with parameters.
TD-31252	Assigning a target schema through the Column Browser does not refresh the page.
TD-31165	Job results are incorrect when a sample is collected and then the last transform step is undone.
TD-30979	Transformation job on wide dataset fails on Spark 2.2 and earlier due to exceeding Java JVM limit. For details, see *https://issues.apache.org/jira/browse/SPARK-18016*.
TD-30857	Matching file path patterns in a large directory can be very slow, especially if using multiple patterns in a single dataset with parameters.
	NOTE: To increase matching speed, avoid wildcards in top-level directories and be as specific as possible with your wildcards and patterns.
TD-30854	When creating a new dataset from the Export Results window from a CSV dataset with Snappy compression, the resulting dataset is empty when loaded in the Transformer page.
TD-30820	Some string comparison functions process leading spaces differently when executed on the Trifacta Photon or the Spark running environment.
TD-30717	No validation is performed for Redshift or SQL DW connections or permissions prior to job execution. Jobs are queued and then fail.
TD-27860	When the platform is restarted or an HA failover state is reached, any running jobs are stuck forever In Progress.

New Known Issues

Ticket	Component	Description
TD-35714	Installer/Upgrader/Utilities	After installing on Ubuntu 16.04 (Xenial), platform may fail to start with "ImportError: No module named pkg_resources" error. **Workaround:** Verify installation of `python-setuptools` package. Install if missing.
TD-35644	Compilation/Execution	Extractpatterns for "HTTP Query strings" option doesn't work.

TD-35562	Compilation/Execution	When executing Spark 2.3.0 jobs on S3-based datasets, jobs may fail due to a known incompatibility between HTTPClient: 4.5.x and aws-java-jdk:1.10.xx. For details, see *https://github.com/apache/incubator-druid /issues/4456* . **Workaround:** Use Spark 2.1.0 instead. In Admin Settings page, configure the `spark. version` property to `2.1.0`. For more information, see *Admin Settings Page*. For additional details on Spark versions, see *Configure for Spark*.
TD-35504	Compilation/Execution	Clicking Cancel Job button generates a 405 status code error. Click Yes button fails to close the dialog. **Workaround:** After you have clicked the Yes button once, you can click the No button. The job is removed from the page.
TD-35486	Compilation/Execution	Spark jobs fail on LCM function that uses negative numbers as inputs. **Workaround:** If you wrap the negative number input in the ABS function, the LCM function may be computed. You may have to manually check if a negative value for the LCM output is applicable.
TD-35483	Compilation/Execution	Differences in how WEEKNUM function is calculated in Trifacta Photon and Spark running environments, due to the underlying frameworks on which the environments are created. • **Trifacta Photon week 1 of the year:** The week that contains January 1. • **Spark week 1 of the year:** The week that contains at least four days in the specified year. For more information, see *WEEKNUM Function*.
TD-35478	Compilation/Execution	The Spark running environment does not support use of multi-character delimiters for CSV outputs. For more information on this issue, see *https://issues.apache.org/jira/browse /SPARK-24540* . **Workaround:** You can switch your job to a different running environment or use single-character delimiters.

TD-34840	Transformer Page	Platform fails to provide suggestions for transformations when selecting keys from an object with many of them.
TD-34119	Compilation/Execution	WASB job fails when publishing two successive appends.
TD-30855	Publish	Creating dataset from Parquet-only output results in "Dataset creation failed" error. **NOTE:** If you generate results in Parquet format only, you cannot create a dataset from it, even if the Create button is present.
TD-30828	Publish	You cannot publish ad-hoc results for a job when another publishing job is in progress for the same job. **Workaround:** Please wait until the previous job has been published before retrying to publish the failing job.
TD-27933	Connectivity	For multi-file imports lacking a newline in the final record of a file, this final record may be merged with the first one in the next file and then dropped in the Trifacta Photon running environment. **Workaround:** Verify that you have inserted a new line at the end of every file-based source.

Release Notes 5.0

Contents:

Release 5.0.1

This release includes a number of key bug fixes and updates.

What's New

- Promote users to Trifacta Administrator role. See *Create Admin Account*.

Changes to System Behavior

None.

Key Bug Fixes

Ticket	Description
TD-31581	Editing joins in reconvergent flows fails with an error message.
TD-31509	Undo not persisted back to server after sample has been collected and loaded.
TD-31399	Join "select-all" performance is slower and can cause browser to hang.
TD-31327	Unable to save dataset sourced from multi-line custom SQL on dataset with parameters.
TD-31305	Copying a flow invalidates the samples in the new copy. Copying or moving a node within a flow invalidates the node's samples.
TD-31165	Job results are incorrect when a sample is collected and then the last transform step is undone.

Security Fixes

The following security-related fixes were completed in this release.

Ticket	Description
TD-33512	In Apache Log4j 2.x before 2.8.2, when using the TCP socket server or UDP socket server to receive serialized log events from another application, a specially crafted binary payload can be sent that, when deserialized, can execute arbitrary code. See *CVE-2017-5645*.
TD-32712	Upgrade Apache portable runtime to latest version to address security vulnerability.
TD-32711	Upgrade Python version to address security vulnerability.
TD-32696	Multiple integer overflows in libgfortran might allow remote attackers to execute arbitrary code or cause a denial of service (Fortran application crash) via vectors related to array allocation. See *CVE-2014-5044*.
TD-32629	Hawk before 3.1.3 and 4.x before 4.1.1 allow remote attackers to cause a denial of service (CPU consumption or partial outage) via a long (1) header or (2) URI that is matched against an improper regular expression. Upgrade version of less to address this security vulnerability. See *CVE-2016-2515*.

TD-32623	Spring Security (Spring Security 4.1.x before 4.1.5, 4.2.x before 4.2.4, and 5.0.x before 5.0.1; and Spring Framework 4.3.x before 4.3.14 and 5.0.x before 5.0.3) does not consider URL path parameters when processing security constraints. By adding a URL path parameter with special encodings, an attacker may be able to bypass a security constraint. The root cause of this issue is a lack of clarity regarding the handling of path parameters in the Servlet Specification. Some Servlet containers include path parameters in the value returned for getPathInfo() and some do not. Spring Security uses the value returned by getPathInfo() as part of the process of mapping requests to security constraints. In this particular attack, different character encodings used in path parameters allows secured Spring MVC static resource URLs to be bypassed. See *CVE-2018-1199*.
TD-32622	Apache POI in versions prior to release 3.15 allows remote attackers to cause a denial of service (CPU consumption) via a specially crafted OOXML file, aka an XML Entity Expansion (XEE) attack. See *CVE-2017-5644*.
TD-32621	math.js before 3.17.0 had an arbitrary code execution in the JavaScript engine. Creating a typed function with JavaScript code in the name could result arbitrary execution. See *CVE-2017-1001002*.
TD-32577	If a user of Commons-Email (typically an application programmer) passes unvalidated input as the so-called "Bounce Address", and that input contains line-breaks, then the email details (recipients, contents, etc.) might be manipulated. Mitigation: Users should upgrade to Commons-Email 1.5. You can mitigate this vulnerability for older versions of Commons Email by stripping line-breaks from data, that will be passed to Email.setBounceAddress(String). See *CVE-2018-1294*.
TD-31427	Apache Commons FileUpload before 1.3.3 DiskFileItem File Manipulation Remote Code Execution See *CVE-2016-1000031*.

New Known Issues

Ticket	Component	Description
TD-31627	Transformer Page - Tools	Prefixes added to column names in the Join page are not propagated to subsequent recipe steps that already existed. > **Workaround:** Perform a batch rename of column names in a step after the join. See *Rename Columns*.
TD-30979	Compilation/Execution	Transformation job on wide dataset fails on Spark 2.2 and earlier due to exceeding Java JVM limit. For details, see *https://issues.apache.org/jira/browse /SPARK-18016* .

New Known External Issues

The following issues are sourced from third-party vendors and are impacting the Trifacta platform.

> **NOTE:** For additional details and the latest status, please contact the third-party vendor listed below.

External Ticket Number	3rd Party Vendor	Impacted Trifacta Feature	Description	Trifacta Ticket
Cloudera Issue: OPSAPS-39589	Cloudera	Publishing to Cloudera Navigator	Within the CDH 5.x product line, Cloudera Navigator only supports Spark 1.x. The Trifacta platform requires Spark 2.1 and later. When Spark 2.x jobs are published to Cloudera Navigator, Navigator is unable to detect them, so they are never added to Navigator. For details, see *https://www.cloudera. com/documentation /enterprise/release-notes /topics /cn_rn_known_issues. html#spark* .	TD-22443

Release 5.0

Release 5.0 of Trifacta® Wrangler Enterprise delivers major enhancements to the Transformer page and workspace, starting with the new Home page. Key management capabilities simplify the completion of your projects and management of scheduled job executions. This major release of the platform supports broader connectivity and integration.

Improving user adoption:

The new workspace features a more intuitive design to assist in building your wrangling workflows with a minimum of navigation. From the new Home page, you can quickly access common tasks, such as creating new datasets or flows, monitoring jobs, or revisiting recent work.

> **Tip:** Check out the new onboarding tour, which provides an end-to-end walkthrough of the data wrangling process. Available to all users on first login of the new release.

Significant improvements have been delivered to the core transformation experience. In the Transformer page, you can now search across dozens of pre-populated transformations and functions, which can be modified in the familiar Transform Builder. Use the new Transformer toolbar to build pre-designed transformations from the menu interface.

New for Release 5.0, target matching allows you to import a representation of the final target schema, against which you can compare your work in the Transformer page. Easy-to-understand visual tags show you mismatches between your current recipe and the target you have imported. Click these tags to insert steps that align your columns with their counterparts in the target.

For multi-dataset operations, the new Auto Align feature in the Union tool improves matching capabilities between datasets, and various enhancements to the Join tool improve the experience.

Over 20 new Wrangle functions deliver new Excel-like capabilities to wrangling.

Enterprise operationalization:

Previously a beta feature, relational connectivity is now generally available, which broadens access to more diverse data. Out-of-the-box, the platform now supports more relational connections with others available through custom configuration. From the Run Jobs page, you can now publish directly to Amazon Redshift.

Build dynamic datasets with variables and parameters. Through parameters, you can apply rules to match multiple files through one platform object, a dataset with parameters. Rules can contain regular expressions, patterns, wildcards, dates, and variables, which can be overridden during runtime job execution through the UI or API. Variables can also be applied to custom SQL datasets.

Using these parameterized datasets allows schedules to pick up new data each execution run and enables users to pass variable values through the API or UI to select different data apply to the job.

Cloud focus:

Release 5.0 delivers broader and enhanced integration with Microsoft Azure. With a few clicks in the Azure Marketplace, you can deploy the platform into a new or existing HDI cluster. Your deployment can seamlessly integrate with either ADLS or WASB and can be configured to connect to Microsoft SQL Data Warehouse. As needed, integrate with Azure Active Directory for single-sign on simplicity.

What's New

Here's what's new in Release 5.0.

Install:

- Support for CDH 5.14.

  > **NOTE:** Support for CDH 5.11 has been deprecated. See *End of Life and Deprecated Features*.

- Support for Spark 2.2.

  > **NOTE:** By default, the Trifacta platform is configured to use Spark 2.1.0. Depending on your environment, you may be required to change the configuration to Spark 2.2, particularly if you are integrating with an EMR cluster. For more information, see *Configure for Spark*.

Azure:

- Integrate your Microsoft Azure deployment with ADLS and WASB.
    - For more information, see *Enable WASB Access*.
    - For more information, see *Enable ADLS Access*.
- Support for Azure Single Sign On. See *Configure SSO for Azure AD*.
    - Integrate with domain-joined clusters using SSO. See *Configure for HDInsight*.
- Support for read-only and read-write connections to Microsoft SQL DW. See *Configure for Azure*.

Admin:

- Through the application, you can now use Tricheck to check the server requirements and connectivity of the Trifacta node to the connected cluster. See *Admin Settings Page*.

Workspace:

- New Home page and left nav bar allows for more streamlined access to recent flows and jobs, as well as learning resources. See *Home Page*.

> **Tip:** Try the tutorial available from the Home page. See *Home Page*.

- Manage your datasets and references from the new Library page. See *Library Page*.
- In the new Jobs page, you can more easily locate and review all jobs to which you have access.
  - Administrators can view and cancel jobs launched by other users.
  - See *Jobs Page*.

Workflow:

- Use parameterized rules in imported datasets to allow scheduled jobs and API executions to automatically pick up the right input data. See *Overview of Parameterization*.
- Assign a new Target to your recipes to provide guidance during wrangling. See *Overview of RapidTarget*.

Transformer Page:

- Search across dozens of pre-defined transformations. Select one, and the Transform Builder is pre-populated based on the current context in the data grid or column browser.
  - See *Search Panel*.
  - See *Transform Builder*.
- Targets assigned to a recipe appear as column header overlay to assist users in aligning their dataset to match the dataset schema to the target schema. See *Data Grid Panel*.
- Cancel in-progress sampling jobs. See *Samples Panel*.
- New toolbar provides faster access to common transformations and operations. See *Transformer Toolbar*.
- Better intelligence for column matching during union operations. See *Union Page*.
- Numerous functional improvements to the Join page. See *Join Panel*.

Run Job Page:

- Specify Redshift publishing actions as part of the job specification. See *Run Job Page*.

Connectivity:

- Delete unused connections through the application. See *Connections Page*.

Changes to System Behavior

> **NOTE:** If you are upgrading an instance that was integrated with an EMR cluster, the EMR cluster ID must be applied to the Trifacta platform. See *Admin Settings Page*.

> **NOTE:** If you are integrating with an EMR cluster, EMR 5.7 is no longer supported. Please create an EMR 5.11 cluster instead. See *End of Life and Deprecated Features*.

Language:

- The aggregate transform has been removed from the platform. Instead, you can use the pivot transform to accomplish the same tasks. For more information, see *Changes to the Language*.

Key Bug Fixes

Ticket	Description

TD-28930	Delete other columns causes column lineage to be lost and reorders columns.
TD-28573	The Trifacta Photon running environment executes column splits for fixed length columns using byte length, instead of character length. In particular, this issue affects columns containing special characters.
TD-27784	Ubuntu 16 install for Azure: supervisord complains about "missing" Python packages.
TD-26069	The Trifacta Photon running environment evaluates `date(yr, month, 0)` as first date of the previous month. It should return a null value.

New Known Issues

Ticket	Component	Description
TD-31354	Connectivity	When creating Tableau Server connections, the Test Connection button is missing. **Workaround:** Create the connection. Create a very simple dataset with minimal recipe. Run a job on it. From the Export Results window, try to publish to Tableau Server. If you cannot connect to the Tableau Server, try specifying a value for the Site Name in the Export Results window.
TD-31305	Workspace	Copying a flow invalidates the samples in the new copy. Copying or moving a node within a flow invalidates the node's samples. **NOTE:** This issue also applies to flows that were upgraded from a previous release. **Workaround:** Recreate the samples after the move or copy.
TD-31252	Transformer Page - Tools	Assigning a target schema through the Column Browser does not refresh the page. **Workaround:** To update the page, reload the page through the browser.
TD-31165	Compilation/Execution	Job results are incorrect when a sample is collected and then the last transform step is undone. **Workaround:** Recollect a sample after undoing the transform step.

TD-30857	Connectivity	Matching file path patterns in a large directory can be very slow, especially if using multiple patterns in a single dataset with parameters. **Workaround:** To increase matching speed, avoid wildcards in top-level directories and be as specific as possible with your wildcards and patterns.
TD-30854	Compilation/Execution	When creating a new dataset from the Export Results window from a CSV dataset with Snappy compression, the resulting dataset is empty when loaded in the Transformer page. **Workaround:** Re-run the job with Snappy compression disabled. Then, export the new dataset.
TD-30820	Compilation/Execution	Some string comparison functions process leading spaces differently when executed on the Trifacta Photon or the Spark running environment.
TD-30717	Connectivity	No validation is performed for Redshift or SQL DW connections or permissions prior to job execution. Jobs are queued and then fail.
TD-30361	Compilation/Execution	Spark job run on ALDS cluster fails when Snappy compression is applied to the output. **Workaround:** Job execution should work if Snappy compression is installed on the cluster.
TD-30342	Connectivity	No data validation is performed during publication to Redshift or SQL DW.
TD-30139	Connectivity	Redshift: No support via CLI or API for: • creating Redshift connections, • running jobs on data imported from Redshift, • publishing jobs results to Redshift **Workaround:** Please execute these tasks through the application.

TD-30074	Type System	Pre-import preview of Bigint values from Hive or Redshift are incorrect.
		Workaround: The preview is incorrect. When the dataset is imported, the values are accurate.
TD-28663	Compilation/Execution	In reference dataset, UDF from the source dataset is not executed if new recipe contains a join or union step.
		Workaround: Publish the source dataset. In the Export Results window, create a new dataset from the results. Import it as your reference data.
TD-27860	Compilation/Execution	When the platform is restarted or an HA failover state is reached, any running jobs are stuck forever In Progress.

Release Notes 4.2

Contents:

- *Release 4.2.2*
  - *What's New*
  - *Changes to System Behavior*
  - *Key Bug Fixes*
  - *New Known Issues*
- *Release 4.2.1*
  - *What's New*
  - *Changes to System Behavior*
  - *Key Bug Fixes*
  - *New Known Issues*
- *Release 4.2*
  - *What's New*
  - *Changes to System Behavior*
  - *Key Bug Fixes*
  - *New Known Issues*

Release 4.2.2

This release includes bug fixes that were previously published as part of Hot Fixes for Release 4.2.1.

What's New

No new features.

Changes to System Behavior

SSL connections to Tableau Server are now functioning. See *Create Tableau Server Connections*.

Key Bug Fixes

Ticket	Description
TD-31118	Update Python to 2.7.14
TD-30307	Update Tomcat libraries to 8.5.28
TD-30239	Tableau publish doesn't allow non-default site and project names with uppercase or spaces.
TD-30210	TDE generation does not include data when the job is run on Spark.
TD-29954	Tableau SSL support. See *Create Tableau Server Connections*.
TD-29777	Dataset owner cannot edit custom SQL.
TD-29562	Cannot change user's upload folder
TD-29552	Data Service is running out of memory
TD-29525	Validation error for Decimal data type when publishing to Hive table
TD-29514	When creating a dataset from a Redshift connection, multiple copies of a column are present.
TD-29387	Spark job runner in batch job runner should not eat exceptions.
TD-29284, TD-29478	Cannot remove structure for shared datasets
TD-29223	Machine learning service fails to gracefully handle parallel requests.
TD-29055, TD-29563	Status Code 403 Forbidden when loading samples
TD-28510	Snapshots containing reused nodes contain multiple copies of each node.
TD-28090	Transformation engine crashes (error -1) when sample validation fails.

New Known Issues

None.

Release 4.2.1

This release includes numerous bug fixes, support for new distributions, and new capabilities, such as the option to disable initial type inference on schematized sources.

What's New

Import:

- Enable or disable initial type inference for schematized sources at global or individual connection level, or for individual dataset sources. See *Configure Type Inference*.

Publishing:

- Support for publishing Datetime data to Hive Datetime or Timestamp data types. See *Hive Data Type Conversions*.

Install, Config & Admin:

- Support for Ubuntu 16.04. See *System Requirements*.

- Support for Cloudera 5.13. See *Supported Deployment Scenarios for Cloudera*.

> **NOTE:** Support for CDH 5.10 has been deprecated. Please upgrade your Hadoop cluster. For more information, see *End of Life and Deprecated Features*.

Changes to System Behavior

None.

Key Bug Fixes

Ticket	Description
TD-27799	DATEDIF function does not work for inputs that are functions returning date values.
TD-27703	Spark job fails with scala.MatchError
TD-24121	When publishing multi-part files, different permissions are written to the parent directory when job was executed on Hadoop or the Tri facta Photon running environment.

New Known Issues

Ticket	Component	Description
TD-27950	Transformer Page - Tools	When you join with an imported dataset not in your flow and it takes longer than expected to collect its initial sample, you may encounter the following error: `Cannot join. Dataset is broken` **Workaround:** Create a recipe off of the imported dataset and then join to the recipe, which is the preferred method of joining. For more information, see *Join Panel*.
TD-27784	Installer/Upgrader/Utilities	Ubuntu 16 install for Azure: supervisord complains about "missing" Python packages. **Workaround:** These packages are present but lack appropriate permissions. A workaround is documented as part of the installation and configuration process. For more information, see "Workaround for missing Python packages," see *Configure for Azure* .

Release 4.2

This release introduces deployment management, which enables separation of development and production flows and their related jobs. Develop your flows in a Dev environment and, when ready, push to Prod, where they can be versioned and triggered for production execution. Additionally, you can create and manage all of your connections through the new Connections page. A revamped flow view streamlines object interactions and now supports starting and stopping of jobs without leaving flow view.

- Release 4.2 also supports installation of the platform on Amazon EC2 instances and integration with EMR as well as installation for Microsoft Azure.

Details are below.

What's New

Deployment Management:

- Manage the lifecycle process of flows across multiple platform instances, building in Dev and publishing to Prod. See *Overview of Deployment Manager*.
- Manage versions deployed into Production. See *Deployment Manager Page*.

Workspace:

- New objects in Flow View and better organization of them. See *Flow View Page*.

> **NOTE:** Wrangled datasets are no longer objects in the Trifacta platform. Their functionality has been moved to other and new objects. For more information, see *Changes to the Object Model*.

  See *Object Overview*.
- Create, manage, and share connections through the new Connections page. See *Connections Page*.
  - Sharing of connections and flows is enabled by default. See *Configure Sharing*.
- Import and export flows from your platform instance.
  - See *Export Flow*.
  - See *Import Flow*.
- Cancel jobs in progress.
  - See *Flow View Page*.
  - See *Jobs Page*.

Transformer Page:

- Perform cross joins between datasets. See *Join Panel*.
- Cut, copy, and paste columns and column values. See *Column Browser Panel*.
- Rename multiple columns in a single transformation step. See *Rename Columns*.
- In Column Details, you can select a phone number or date pattern to generate suggestions for standardizing the values in the column to a single format. See *Column Details Panel*.

Personalization:

- Personalized suggestions presented based on your previous usage.
- Browse and select patterns for re-use from your recent history. See *Pattern History Panel*.
- Upload your own avatar image. See *User Profile Page*.

> **NOTE:** This feature may need to be enabled. See *Miscellaneous Configuration*.

Install/Admin/Config:

- Install from Amazon Marketplace via AMI into a deployed EC2 instance.
- Leverage IAM roles to manage permissions for the Trifacta platform deployed on an EC2 instance. See *Configure for EC2 Role-Based Authentication*.
- Install and integrate with Amazon Elastic MapReduce (EMR). See *Configure for EMR*.
- Install for Microsoft Azure and integrate with HDInsight. See *Install from Azure Marketplace*.

Integration:

- Redshift improvements:
  - The Trifacta platform supports multiple private and global connections to Redshift databases. See *Create Redshift Connections*.
  - You can read from Redshift databases. See *Redshift Browser*.
- Publish directly to Tableau Server. See *Run Job Page*.
  - For more information on creating the connection, see *Create Tableau Server Connections*.

Language:

- New string comparison functions.
- New SUBSTITUTE function replaces string literals or patterns with a new literal or column value.
- See *Changes to the Language*.

Import:

- Expanded set of encoding types supported for file import. See *Configure Global File Encoding Type*.

Performance:

- Improved performance when initializing jobs and in Flow View for complex flows.

Changes to System Behavior

New session duration parameter and default value

For technical reasons, the name and default value of the following parameter has been changed in Release 4.2.

Affected Releases	Parameter Name	Default Value	Max Value
Release 4.2 and later	`webapp.session.DurationInMins`	10080 (one week)	30000
Release 4.1.1 and earlier	`webapp.session.DurationInMinutes`	43200 (one month)	30000

> **NOTE:** Upgrading customers have the new configuration setting automatically set to the default: `10080` m inutes (one week). You must make adjustments as needed.

For more information on changing this parameter value, see *Configure Application Limits*.

/docs endpoint is removed

In Release 4.0, the `/docs` endpoint was deprecated from use. This endpoint displayed a documentation page containing information on Wrangle language, the command line interface, and Trifacta patterns.

In Release 4.2, this endpoint has been removed from the platform. Content has been superseded by the following content:

- See *Wrangle Language*.
- See *Command Line Interface*.
- See *Text Matching*.

For more information on features that have been deprecated or removed, see
End of Life and Deprecated Features.

s3n is no longer supported

If you are integrating with S3 sources, the platform now requires use of the s3a protocol. The s3n protocol is no longer supported.

No configuration changes in the Trifacta platform are needed. See *Enable S3 Access*.

Key Bug Fixes

Ticket	Description
TD-27748	Direct publish to Hive fails on wide datasets due to Avro limitations.
TD-27368	SQL Server Database timing out with long load times. • For more information on timeout settings, see *Configure Application Limits*.
TD-27197	Column histogram does not update after adding `pluck` parameter to `unnest` transform.
TD-27127	Send a Copy tab in Flow View sharing does not include all available users.
TD-27055	Job run on flow with complex recipes fails on Hadoop but succeeds on the Trifacta Photon running environment.
TD-26837	Creating custom dictionaries fails on S3 backend datastore.
TD-26388	Orphaned bzip2 processes owned by the platform user accumulate on the node.
TD-26041	When editing a schedule that was set for 0 minutes after the hour, the schedule is displayed to execute at 15 minutes after the hour.
TD-25903	Overflow error when ROUND function is applied to large values.
TD-25733	Attempting a union of 12 datasets crashes UI.
TD-25709	Spark jobs fail if HDFS path includes commas.

New Known Issues

Ticket	Component	Description
TD-27799	Compilation/Execution	DATEDIF function does not work for inputs that are functions returning date values. **Workaround:** Write function returning your date values to a new column. Then, apply DATEDIF function using that column as a new input.
TD-27703	Compilation/Execution	Spark job fails with scala.MatchError
TD-26069	Compilation/Execution	The Trifacta Photon running environment evaluates `date(yr, month, 0)` as first date of the previous month. It should return a null value.

TD-24121	Compilation/Execution	When publishing multi-part files, different permissions are written to the parent directory when job was executed on Hadoop or the Trifacta Photon running environment.

Release Notes 4.1

Contents:

- *Release 4.1.1*
  - *What's New*
  - *Changes to System Behavior*
  - *Key Bug Fixes*
  - *New Known Issues*
- *Release 4.1*
  - *What's New*
  - *Changes to System Behavior*
  - *Key Bug Fixes*
  - *New Known Issues*

Release 4.1.1

This release of Trifacta® Wrangler Enterprise introduces scheduling of dataset execution from within your flows, as well as a number of bug fixes and system improvements.

What's New

Admin, Install, & Config:

- Support for Cloudera 5.12. See *Supported Deployment Scenarios for Cloudera*.

> **NOTE:** Support for Cloudera 5.9 has been deprecated. For more information, see *End of Life and Deprecated Features*.

Workspace:

- Schedule executions of one or more wrangled datasets within a flow. See *Flow View Page*.

Transformer Page:

- Disable individual steps in your recipes. See *Recipe Panel*.
- Search for columns by name. See *Data Grid Panel*.

Changes to System Behavior

Single-file run_job action is deprecated for CLI

See *Changes to the Command Line Interface*.

Key Bug Fixes

Ticket	Description
TD-25615	Error in flat-aggregation generation in Spark running environment.
TD-25438	Deleting an upstream reference node does not propagate results correctly to the Transformer page.
TD-15509	TDE files generated by the TDE download option may fail to open in Tableau if column names are more than 124 characters in length. **NOTE:** When you run the job, include a publishing option to publish to TDE format. When you export the generated results, this issue no longer appears in the output.

New Known Issues

Ticket	Component	Description
TD-26041	Workspace	When editing a schedule that was set for 0 minutes after the hour, the schedule is displayed to execute at 15 minutes after the hour. **Workaround:** This bug is a display bug. The correct value is saved when the value is set to 0 for the schedule.

Release 4.1

This release of Trifacta Wrangler Enterprise includes the ability to share flows and a completely revamped Transformer page for a simpler, faster, and more consistent user experience. From the Transformer page, you can now collect ad-hoc samples using a wider variety of techniques. New integration and publishing options make the Trifacta platform broader in its reach throughout the enterprise. Read below for additional features and details.

What's New

Admin, Install, & Config:

> **Support for integration with MapR Hadoop clusters has been deprecated. The Trifacta platform continues to support Cloudera and Hortonworks. For more information on other available options, please contact your Trifacta representative.**

> **NOTE:** Support for CentOS 6.2.x and CentOS 6.3.x has been deprecated. Please upgrade to the latest CentOS 6.x release.

- Support for Cloudera 5.11. See *Supported Deployment Scenarios for Cloudera*.

> **NOTE:** Support for CDH 5.8 has been deprecated. See *End of Life and Deprecated Features*.

- Support for HDP 2.6. See *Supported Deployment Scenarios for Hortonworks*.

> **NOTE:** Support for HDP 2.4 has been deprecated. See *End of Life and Deprecated Features*.

- Integration with Alation data catalog service. See *Enable Alation Sources*.
- Integration with Waterline data catalog service. See *Enable Waterline Sources*.

Import:

- Support for large-scale relational sources when executing jobs on Hadoop. See *Enable Relational Connections*.
- Per-file import settings including file encoding type, automatic structure detection. See *Import Data Page*.

> **NOTE:** The list of supported encoding types has changed. See *Configure Global File Encoding Type*.

- Read/write support for Snappy compression. See *Supported File Formats*.

> **NOTE:** Integration with fully compressed Hadoop clusters requires additional configuration. See *Enable Integration with Compressed Clusters*.

Workspace:

- Improved user experience with flows. See *Flow View Page*.
- Share a flow with one or more users, so you can collaborate on the same assets. See *Flow View Page*.

Transformer Page:

- New navigation and layout for the Transformer page simplifies working with data and increases the area of the data grid. See *Transformer Page*.
- Sampling improvements:
    - Enhanced sampling methods provide access to customizable, task-oriented subsets of your data. See *Samples Panel*.
    - Improved Transformer loading due to persistence of initial sample.
    - For more information on the new sampling methods, see *Overview of Sampling*.
- Highlight the recipe steps where a specific column is referenced. See *Column Menus*.

Compilation/Execution:

- Publishing to Hive:
    - You can now publish directly to Hive as part of job execution. Just configure a new publishing action. See *Run Job Page*.
    - Enhanced publishing options for Hive target tables including Create, Append, Drop & Truncate. See *Publishing Dialog*.
- Trifacta Photon jobs can be automatically killed based on configurable runtime and memory consumption thresholds. See *Configure Photon Running Environment*.
- The Trifacta Photon running environment now supports Parquet format.

Admin:

- SSO integration with AD/LDAP now supports auto-registration for users visiting the Trifacta application. See *Configure SSO for AD-LDAP*.

Language:

- New `CASE` function.
- For more information, see *Changes to the Language*.

Changes to System Behavior

Hadoop Pig running environment is no longer available

As of Release 4.1, the Pig running environment is no longer available for execution of jobs. Implications:

- Deployments that are connected to a Hadoop cluster must use Spark for job execution. See *Configure Spark Running Environment*.
- CLI scripts that reference running jobs on the `pig` running environment must be updated. See *Changes to the Command Line Interface*.
- Integration with Cloudera Navigator is not supported in this release.
- Integration with HDI/WASB is supported but may require further configuration. Please contact *Trifacta Support*.

Python UDFs are no longer available

With the removal of the Hadoop Pig running environment, Python user-defined functions are no longer available.

> **NOTE:** As of Release 4.1, all user-defined functions must be migrated to or created in Java. For more information, see *Java UDFs*.

For more information on migrating UDFs, see *Changes to the User-Defined Functions*.

Transform Editor has been removed

In Release 4.0.1 and earlier, you could type in Wrangle transformation steps as plain text in the Transform Editor as well as use the Transform Builder.

In Release 4.1 and later, the Transform Editor has been removed, in favor of an enhanced version of the Transform Builder.

> **Tip:** You can copy and paste raw Wrangle commands into the Transformation/Choose a transformation textbox of the Transform Builder. The documentation still displays example transformation steps as Wrangle text commands.

See *Transform Builder*.

Dependencies Browser has been replaced

In Release 4.0.1, you could explore dependencies between your datasets through the Dependencies Browser, which was accessible through a graph in the toolbar in the Transformer page.

In Release 4.1, this browser has been replaced by the Dataset Navigator. In the Transformer page, click the drop-down next to the name of the current dataset. In the Dataset Navigator, you can browse the datasets through a list or flow view to locate another wrangled dataset to load.

In Release 4.2 and later, this browser has been renamed to the Recipe Navigator. See *Recipe Navigator*.

Manual database installation is no longer required

Prior to Release 4.0, the databases had to be installed manually.

In Release 4.0 and later, the databases are installed for you on the local server as part of the basic install process. For more information, see *Install Databases*.

If you need to re-install the databases, manual steps are still available. See *Install Databases for PostgreSQL*.

Head sample replaced by random sample on upgrade

In Release 4.0 and earlier, if your dataset used the initial rows (head) sample in the data grid, this sample is replaced by the random sample after the upgrade.

> **Tip:** When the dataset is loaded in the Transformer page after upgrade, you can switch the sample back to the first rows sample. For more information, see *Samples Panel*.

Miscellaneous

- The Send a Copy feature introduced in Release 4.0 has been integrated with the general sharing capabilities. See *Share Flow Dialog*.
- Ah-hoc publishing to Redshift in CSV format is no longer supported. See *Publishing Dialog*.

Key Bug Fixes

Ticket	Description
TD-23787	When publishing location is unavailable, spinning wheel hangs indefinitely without any error message.
TD-22467	Last active sample is not displayed during preview of multi-dataset operations.
TD-22128	Cannot read multi-file Avro stream if data is greater than 500 KB.
TD-20796	For date column, Spark profiling shows incorrect set of dates when source data has a single date in it.
TD-19865	You cannot configure a publishing location to be a directory that does not already exist. See *Run Job Page*.
TD-17657	`splitrows` transform allows splitting even if required parameter `on` is set to an empty value.

New Known Issues

Ticket	Component	Description
TD-25419	Profiling	When a pivot transform is applied, some column histograms may not be updated. **Workaround:** Refresh the page.

TD-25000	Connectivity	Cannot publish to Cloudera Navigator due to 500 - Internal Server Error.
		Workaround: The Cloudera Navigator integration is not supported in this release. If it has been enabled in your deployment in a prior release, it must be disabled. To disable, please set the following property value in platform configuration. You can apply this change through the *Admin Settings Page* (recommended) or `trifacta -conf.json`. For more information, see *Platform Configuration Methods*.
TD-24358	Compilation/Execution	Circular reference in schema of Avro file causes job in Spark running environment to fail. See *https://issues.apache.org/jira/browse /AVRO-1285* .
TD-20882	Connectivity	Spark jobs based on relational sources fail if one or more columns is dropped from the source table.
TD-21836	Transformer Page	Values in a newly collected sample do not appear in sorted order, even though a sort transform had been previously applied.
		Workaround: You can re-apply the sort transform to the new sample. Some limitations apply. For more information, see *Sort Transform*.

Release Notes 4.0

Contents:

Release 4.0.2

This release contains key bug fixes from Release 4.0.1.

What's New

No new features have been introduced.

Changes to System Behavior

None.

Key Bug Fixes

Ticket	Description
TD-25182	Update NodeJS to 6.11.1
TD-25143	Spark job gets stuck for flow with header filter and multiple map transform expressions
TD-25090	Spark job OOM error when failing over frequently on a Resource Manager High Availability cluster
TD-25087	Dictionary URL is incorrect in CDF for Spark jobs
TD-25080	Spark jobs with timestamp source columns yield empty columns
TD-24965	Job fails with "Unary operator LexiconCheck not supported" in Spark
TD-24869	Corrupted `DotZlib.chm` file in 4.0.1 RPM
TD-24669	Nginx Request URI length default is too low.
TD-24464	'Python Error' when opening recipe with large number of columns and a nest
TD-24409	ArrayIndexOutOfBoundsException when UDF iterator reaches premature end
TD-24322	Nest transform creates a map with duplication keys.
TD-23921	In shared Hadoop cluster on Edge environment, valid relational connections do not appear in the GUI.
TD-23920	Support for equals sign (=) in output path.
TD-23904	Results of Spark job show missing values, even though recipe step replaces them with a value.
TD-23857	Type registry fails to initialize when webapp process is relaunched.

TD-23791	Spark PyMultiStringReplaceUdf UDF code throws NPE when processing nested fields.
TD-23780	Unexpected dates appear in CSV output on Trifacta Photon job execution.
TD-23722	umask settings on output directories are not being respected for single-file output.
TD-23646	Adding a specific comment appears to invalidate earlier edit.
TD-23645	Spark unable to read recursive folders
TD-23578	Spark error doing split
TD-23507	No rows in random samples on CSM cluster.
TD-23459	Recipe upgraded from 3.1 to 3.2 becomes corrupted when new lookup is added.
TD-23457	Webapp, batch-job-runner scaling issues
TD-23358	Flow with many dependencies hangs for 6 hours and then fails when executed in Spark on AWS
TD-23276	Generating large CLI script blocks client access
TD-23111	Long latency when loading complex flow views
TD-23102	Recipe showing MISSING for some Lookups after upgrade
TD-23099	View Results button is missing on Job Cards even with profiling enabled
TD-22907	Spark yarn-app log dump feature requires Trifacta account to have read/execute permissions to log aggregation folder.
TD-22889	Extremely slow UI performance for some actions
TD-22796	Java UDFs must support initSchema method to initArgs.
TD-22313	Use Node.js cluster module for easy scaling of webapp and VFS services
TD-22291	Columns created from UDFs do not work with column browser, column menus, or both, and they cannot be shown or hidden.

New Known Issues

None.

Release 4.0.1

This release adds a few new features and addresses some known issues with the platform.

What's New

Admin, Install, & Config:

> **NOTE:** Integration with MapR is not supported for this release.

- Support for Cloudera 5.10. See *Supported Deployment Scenarios for Cloudera*.
- Access to S3 buckets can now be controlled on a per-user basis. See *Enable S3 Access*.
- More parameters now available through the application. See *Admin Settings Page*.
- Send Spark jobs to a specified YARN queue. See *Configure for Spark*.

- You can now configure the default file format for jobs run on the Hadoop cluster. See *Configure for Hadoop*.
  - Different file formats and other options can still be configured as part of the job. See *Run Job Page*.
- Support for CentOS/RedHat Linux 7.1 - 7.x on Trifacta node. See *System Requirements*.

Language:

- Apply optional `quoteEscapeChar` to identify escaped quote characters when splitting rows.
- See *Changes to the Language*.

Changes to System Behavior

Application timeout behavior more consistent

In Release 4.0, the web application session timeout was set to 60 minutes by default, which caused inconsistent behaviors. See TD-22675 below.

In Release 4.0.1 and later, this session timeout was set to one month by default. This change returns the web application to the same setting as Release 3.2.1 and earlier.

> **NOTE:** Beginning in Release 4.0, this setting is configurable. For more information on changing the session timeout, see *Configure Application Limits*.

Key Bug Fixes

Ticket	Description
TD-22675	Session timeout behavior is inconsistent. Application seems to have some functionality after timeout.
TD-22570	After upgrade, some pre-upgrade jobs appear to point to deleted datasets.
TD-22388	S3 authorization mechanism does not support Signature Version 2 in Asia-Pacific and EU.
TD-22220	Dataset suddenly fails to load after upgrade from Release 3.2 because of type checking on an invalid recipe line.
TD-19830	Editing a Join or Union transform that includes a reference dataset (not in the same flow) may result in the unintentional removal of that reference dataset from the flow.
TD-14131	`splitrows` transform does not work after a backslash. This issue is fixed with the new `quoteEscapeChar` parameter for the `splitrows` transform. See *Changes to the Language*.
TD-5783	Prevent two-finger scroll in data grid from stepping back in the browser's history on Mac OS.

New Known Issues

Ticket	Component	Description

TD-22864	Compilation/Execution	Connection for Redshift publishing uses its own AWS access key and secret, which may be different from the per-user or system credentials. If the Redshift connection does not have read access to the data, publication fails.
		Workaround: Verify that the access key and secret for the Redshift connection has access to any source data that you wish to publish to Redshift.

Release 4.0

This release features a single page for managing your flows, a faster Spark-based running environment on the Trifacta node, and a number of new Wrangle functions and capabilities. Details are below.

> **NOTE:** Integration with MapR is not supported for this release.

What's New

Workspace:

- The new flow detail page includes a visual representation of your flow and detailed information about its datasets and recipes. From the Flow View page, users can swap datasets and run jobs, too. See *Flow View Page*.
- Send a copy of a flow to another user. See *Send a Copy of a Flow*.

Transformer Page:

- Column width settings now persist across transform steps, other actions, and user sessions. See *Transformer Page*.
- Users can now perform join and unions directly against imported datasets that contain schema information, such as Hive, JDBC, and Avro.
- Wrangle steps can now be displayed in natural language. See *Data Grid Panel*.
- New column menu shortcuts allow you to quickly assemble recipe steps from menu selections, based on a column's data type. See *Column Menus*.
- New column browser streamlines interactions involving multiple columns. See *Column Browser Panel*.
- Default quick scan samples are now collected over more of the data source, the first 1 GB. Administrators can now modify this size. See *Configure Application Limits*.
- For the Spark running environment, you can enable generation of random samples across the entire dataset. See *Configure for Spark*.

Profiling:

- Enhanced pattern profiling enables streamlined processing of fixed-width datasets. See *Parse Fixed-Width File and Infer Columns*.

Ingestion:

- New Custom SQL query options for Hive and relational sources enables pre-filtering of rows and columns by executing the SQL logic within the database to reduce data transfer time for faster overall performance. See *Enable Custom SQL Query*.
- Users can now import Hive views to be used as a source. See *Hive Browser*.

- Expand the list of file extensions that are permitted for upload. See *Miscellaneous Configuration*.

Compilation/Execution:

- New Spark v2.1.0-based running environment leverages in-memory speed to deliver overall faster execution times on jobs. See *Configure Spark Running Environment*.

> **NOTE:** As of Release 4.0, for new installs and upgrades, Spark is the default running environment for execution on the Hadoop cluster. Support for Hadoop Pig running environment is deprecated and in future releases will reach end-of-life. For more information, see *Running Environment Options*.

> **NOTE:** Python UDFs are not supported in the Spark running environment. Support for Python UDFs is deprecated and in a future release will reach end-of-life. For more information on migrating to using Java UDFs, see *Changes to the User-Defined Functions*.

- You can disable the ability to run jobs on the Trifacta node. See *Running Environment Options*.
- User-specific properties can be passed to Pig or Spark for use during job execution. See *Configure User-Specific Props for Cluster Jobs*.
- Default file publishing setting for CSV output is multiple output files when using a Hadoop running environment, resulting in better performance over large data volumes.

Language:

- Window transform now supports use of aggregation functions. See *Window Transform*.
- New `NOW` and `TODAY` functions.
    - See *NOW Function*.
    - See *TODAY Function*.
- New `ROLLINGSUM` function computes the rolling sum over a specified number of rows before and after the current row. See *ROLLINGSUM Function*.
- New `ROLLINGAVERAGE` function computes rolling average over a specified window. See *ROLLINGAVERAGE Function*.
- New `ROWNUMBER` function computes the row number for each row, based on order and optional grouping parameters. See *ROWNUMBER Function*.
- New `COUNTA` function can be used to count the number of non-null values in a column based on order and grouping parameters. See *COUNTA Function*.
- New `COUNTDISTINCT` function counts distinct number of values in a specified column. See *COUNTDISTINCT Function*.
- Four new functions for testing conditional data validation: `IFNULL`, `IFMISMATCHED`, `IFMISSING,` and `IFVALID`. See *Type Functions*.
- New `*IF` functions for each available aggregation function. See *Aggregate Functions*.
- For more information, see *Changes to the Language*.

APIs:

- First release of publicly available APIs, which enable end-to-end operationalization of processing your datasets. See *API Reference*.

CLI:

- Add custom properties to your jobs when executing via CLI on the Hadoop cluster (i.e. YARN queue) See *Configure User-Specific Props for Cluster Jobs*.

Admin, Install, & Config:

- Support for HDP 2.5. See *Supported Deployment Scenarios for Hortonworks*.
- Support for non-default users and groups. See *Required Users and Groups*.

- New Admin Settings page exposes all platform configuration that is available through the application for easy search, updating, and validation. See *Changes to the Admin Settings Page*.
- Configurable log levels for key platform services. See *Configure Logging for Services*.
- Pre-upgrade samples are now persisted after upgrade is complete.
- Trifacta administrators can download services logs through the application, instead of the Trifacta node. See *System Services and Logs*.

Changes in System Behavior

Changes to the Language:

- `set` and `settype` transforms now work on multiple columns.
- Recipe steps are now displayed in natural language format by default in the recipe panel and suggestion cards.
- Some functions have been renamed to conform to common function names.
- For more information, see *Changes to the Language*.

Changes to the CLI:

- The Jobs command line interface now supports job execution on the Spark running environment. See *CLI for Jobs*.

End of Life Features:

- The Javascript running environment and profiler are no longer supported. Use the Trifacta Photon running environment instead. For more information, see *Running Environment Options*.
- The Hadoop Pig profiler and the Python-based Spark profiler are no longer supported. Use the Scala profiler instead. See *Profiling Options*.
- The `/docs` for inline documentation is no longer supported. Content in that location has been replaced and superseded by content in product documentation.
    - See *Command Line Interface*.
    - See *Wrangle Language*.
    - See *Text Matching*.
- For more information, see *End of Life and Deprecated Features*.

Key Bug Fixes

Ticket	Description
TD-21006	The Trifacta Photon running environment fails to compress output file and is forced to restart on download.
TD-20736	Publish to Redshift fails for single-file outputs.
TD-20524	Join tool hangs due to mismatched data types.
TD-20344	When the Trifacta Photon client is enabled, no sample data is displayed when joins yield a data mismatch.
TD-20176	After Release 3.2.1 upgrade, data grid in the Transformer Page no longer displays any data in the sample, even though data is present in the pre-upgrade environment.
TD-20173	`NUMFORMAT` string `#.#0` fails to be converted to supported string format on upgrade, and recipe step fails validation. For more information, see *Changes to the Language*.
TD-19899	Failed first job of jobgroup prevents datasets from showing up in flow.
TD-19852	User can accept compressed formats for append publish action.
TD-19678	Column browser does not recognize when you place a checkmark next to the last column in the list.

TD-18836	`find` function accepts negative values for the start index. These values are consumed but produce unexpected results.
TD-18746	When the Trifacta Photon client is enabled, previews in the data grid may take up to 30 seconds to dismiss.
TD-18538	Platform fails to start if Trifacta user for S3 access does not have the ListAllMyBuckets permission.
TD-18340	When writing CSV outputs, the Spark running environment fails to recognize the defined escape character.
TD-17677	Remove references to Zookeeper in the platform.
TD-16419	Comparison functions added through Builder are changed to operators in recipe
TD-12283	Platform cannot execute jobs on Pig that are sourced from S3, if OpenJDK is installed.

New Known Issues

Ticket	Component	Description
TD-22128	Complication/Execution	Cannot read multi-file Avro stream if data is greater than 500 KB. **Workaround:** Load files as independent datasets and union them together, or concatenate the files outside of the platform.
TD-21737	Transformer Page	Cannot transform downstream datasets if an upstream dataset fails to contain a `splitrows` transform. **Workaround:** Add a `splitrows` transform to the upstream dataset. See *Splitrows Transform*.
TD-20796	Job Results Page	For date column, Spark profiling shows incorrect set of dates when source data has a single date in it.

TD-19183	Workspace	Merge function does not work with double-escaped values, and job fails in Pig. Example:

```
set col:
column4 value:
merge(['ms\\',
column4])
```

Workaround: Add a dummy character to the original transform and then remove it. Example:

```
set col:
column4
value:
merge
(['ms\\',
column4])
replace
col:
column4
on: ''
with: ''
```

As another alternative, you can execute the job in the Spark running environment.

Release Notes 3.2

Contents:

Release 3.2.1

This release contains numerous bug fixes and some interesting new features.

What's New

Transformer Page:

- Specify create, append, or replace actions for your file publishing destinations. See *Run Job Page*.

Workspace:

- Import to Wrangle in one step. See *Import Data Page*.

Admin, Install, & Config:

- Support for CDH 5.9. See *Supported Deployment Scenarios for Cloudera* .

> **NOTE:** Support for CDH 5.5/CDH 5.6 has been deprecated. Please upgrade to CDH 5.8 or later.

Changes to System Behavior

Changes to the Language:

- A number of functions have been renamed to conform to common function names. See *Changes to the Language*.

Changes to the Command Line Interface:

- New file publishing options enable specifying create, append, and replace actions for file publishing destinations.
  - `output_path` is now a required parameter for commands that use it.

> **NOTE:** When specifying publishing options in the CLI, you may specify one file format only f or the output.

- See *Changes to the Command Line Interface*.

Miscellaneous Changes:

- The setting to include headers in CSV downloads is now managed as part of the job publication workflow on a per-job basis.
  - This setting is no longer available in the Admin Settings page.
  - For more information, see *Run Job Page*.
- Access to S3 sources no longer requires the ListAllMyBuckets permission. If the permission is not granted:
  - Users cannot see default buckets through the application.
  - Default buckets must be explicitly configured to be displayed from within the application.
  - Users can still access unlisted buckets by directly entering the full path in the S3 browser.
  - See *Enable S3 Access*.

Key Bug Fixes

Ticket	Description
TD-19404	Split transform using at parameter values out of range of cell size generates an error in Pig.
TD-19150	On Photon, `unnest` transform fails if `pluck=true`.

TD-19032	Swapping rapidly between source datasets that have already been edited may cause a `No samples found` error.
TD-18933	You cannot load a dataset that utilizes another dataset via join or union three levels deep.
TD-18268	If you profile a wide column (one that contains many characters of data in each cell value), the machine learning service can crash.
TD-18093	Changes to a dataset that generates new columns can break any downstream lookups that use the dataset.

New Known Issues

Ticket	Component	Description
TD-20736	Compilation/Execution	Publish to Redshift of single-file CSV or JSON files fails. **Workaround:** Publish files to Redshift as multi-part files. See *Run Job Page*.
TD-19898	Installer/Upgrader/Utilities	After upgrade, job card summaries in the Jobs page may fail to load for jobs executed in the pre-upgrade version with steps containing functions that have been renamed. **Workaround:** You can re-run the job in the upgraded version. For more information on the renamed functions for Release 3.2.1, see *Changes to the Language*.
TD-19870	Compilation/Execution	When publishing to S3, you cannot write to a single file in an `append` publishing action. **Workaround:** You can change the publish action to recreate the object, replace the object, or save it as a multi-file output.
TD-19866	Transformation	When switching between an `append` and a `replace` publishing action in the application, any selected compression selecting is lost. You cannot set this value again. **Workaround:** Cancel the edit in progress. Re-edit the publishing action to apply the compression setting to the `replace` transform.

TD-19865	Workspace	You cannot configure a publishing location to be a directory that does not already exist.
		Workaround: Create the directory on the datastore outside of the Trifacta platform. Verify that the appropriate user accounts have access to the directory.
TD-19852	Compilation/Execution	User are permitted to select compressed formats for `append` publish action, which is not supported.
		NOTE: For Release 3.2.1, the `append` publish action does not support the use of compression.
TD-19827	Compilation/Execution	Job execution fails with `java.lang.OutOfMemoryError: unable to create new native thread` exception in job log.
		Workaround: You can try to raise the soft and hard limit on number of processes available to the platform. For more information, see *Miscellaneous Configuration*.
TD-19678	Transformer Page	Column browser does not recognize when you place a checkmark next to the last column in the list.
		Workaround: You can move the column to another location and then select it.
TD-19384	Transformer Page	Preview cards take a long time to load when selecting values from a Datetime column.
		Workaround: For selection purposes, you can change the data type to String. Then, make your selections and build your transform steps before switching back to Datetime data type.

TD-18584	Type System	`settype` transforms that do not include a specified Datetime formatting string and its variant fail on upgrade. In previous releases, this formatting was permitted, and the variant to apply was inferred. **Workaround:** Please review the variant information in the transform. Then, remove the step and re-apply the Date formatting through the Type drop-down for the column. The required type information is applied.

Release 3.2

This release features the introduction of the following key features:

- A new and improved object model.
- A completely redesigned execution engine (codename: Photon), which enables much better performance across larger samples in the Transformer page and faster execution on the Trifacta Server.

> **NOTE:** To interact with the Photon running environment, all desktop instances of Google Chrome must have the PNaCl component enabled and updated to the minimum supported version. See *Desktop Requirements*.

> **NOTE:** If you are upgrading from Release 3.1.x, you must manually enable the Photon running environment. If you are upgrading from an earlier version or installing Release 3.2 or later, the Photon running environment is enabled by default. See *Configure Photon Running Environment*.

- The Transform Builder, a menu-driven interface for rapidly building transforms.
- A new publishing interface with easier, more flexible configuration.
- Numerous other features and performance enhancements.

Details are below.

What's New

Object Model:

- Redesigned object model and related changes to the Trifacta application enable greater flexibility in asset reuse in current and future releases.

> **NOTE:** Beginning in Release 3.2, the Trifacta platform is transitioning to an enhanced object model, which is designed to support greater re-usability of objects and improved operationalization. This new object model and its related features will be introduced over multiple releases. For more information, see *Changes to the Object Model*.

Transformer Page:

- A newly designed interface helps you to quickly build transform steps. See *Transform Builder*.
- New publishing interface with more flexible configuration options for outputs. See *Run Job Page*.

- Scrolling and loading improvements in the Transformer page.
- Substantial increase in the size of samples in Transformer page for better visibility into source data and more detailed profiling.
- Use the Dependencies Browser to review and resolve dependency errors between your datasets. See *Recipe Navigator*.
  - For more information on the implications, see *Changes to the Object Model*.
- Explore automatically detected string patterns in column data using pattern profiling and build transforms based on these patterns. See *Column Details Panel*.
- Join tool now supports fuzzy join options. See *Join Panel*.

Admin, Install, & Config:

> **NOTE:** The minimum system requirements for the Trifacta node have changed for this release. For more information, see *System Requirements*.

- Support for CentOS/RedHat Linux 7.1. See *System Requirements*.
- Support for Ubuntu 14.04.
  - Ubuntu 12.04 is no longer supported.
  - See *System Requirements*.
- Support for CDH 5.8 core and with security. See *Supported Deployment Scenarios for Cloudera* .

> **NOTE:** Support for CDH 5.3/CDH 5.4 has been deprecated. Please upgrade to CDH 5.8 or later.

- Support for Hortonworks 2.4 with security. See *Supported Deployment Scenarios for Hortonworks*.
- Configurable session duration. See *Miscellaneous Configuration*.
- Support for Google Chrome 51+ only. See *Desktop Requirements*.
- Connect to multiple deployments of the Trifacta Server through the Wrangler Enterprise desktop application . See *Configure for Trifacta Enterprise Application*.

Command Line Interface:

- Support for use of Kerberos credentials by the CLI.
- Support for asset transfer during user deletion.
- See *Changes to the Command Line Interface*.

APIs:

- Support for end-to-end integration via API and CLI. For more information on content, please contact *Trifacta Support*.

Job Execution and Performance:

- Superior performance in job execution. Run jobs on the Trifacta Server on much larger datasets and faster rate.
- Numerous performance improvements to the web application across many users.
- New Batch Job Runner service simplifies job monitoring and improves performance.

> **NOTE:** The Batch Job Runner service requires a separate database for tracking jobs. New and existing customers must manually install this database. See *Install Databases for PostgreSQL*.

- Improved error message on job failure.

Connectivity:

- Publish to Redshift is now generally available. See *Enable S3 Access*.
- SSL support for Oracle, Postgres, and Teradata relational sources. See *Create Connection Window*.

Security:

- Numerous security enhancements.

Changes to System Behavior

This section outlines changes to how the platform behaves that have resulted from features or bug fixes in Release 3.2.

Post-Upgrade Sampling

> **NOTE:** Due to changes in system behavior, all existing random samples for a dataset are no longer available after upgrading to this release. For any upgraded dataset, the selected sample reverts to the default sample, the first N rows of the dataset. The number of rows in the sample depends on the number of columns, data density, and other factors.

When you load your dataset into the Transformer page for the first time:

- The first N rows of the dataset is selected as a sample.

> **NOTE:** The first N rows sample may change the data that is displayed in the data grid. In some cases, the data grid may initially display no data at all.

- A new random sample is automatically generated for you.
- The Collect New Random Sample button is available. However, until you add a script step that changes the number of rows in the dataset, this button creates a random sample that is identical to the one that is automatically created for you when you first load the dataset into the Transformer page.

Changes to Wrangle

- The `multisplit` transform has been replaced by a more flexible version of the `split` transform. For more information, see *Split Transform*.
- Additional miscellaneous changes. See *Changes to the Language*.

Key Bug Fixes

Ticket	Description
TD-18319	Inconsistent results for `DATEDIFF` function across running environments. For more information, see *Changes to the Language*.
TD-16255	`windowfill` function fails to fill over some empty cells.
TD-16086	Job list drop-down fails to enable selection of correct jobs.
TD-16084	Job cards display `CLI Job` source for jobs launched from the application.
TD-15609	Column filtering only works if filtering value is entered in lowercase.
TD-15442	Attempt to publish to Cloudera Navigator for a Trifacta® Server job results in a DataNotFoundException.
TD-15330	Pivot transform generates "Cannot read property 'primitive' of undefined" error.

TD-14541	Names for private connections can collide with names of global connections, resulting in private connection unable to be edited by the owning user.
TD-14397	Left or outer join against dataset with `deduplicate` as last script line fails in Pig execution.
TD-13162	Join key selection screen and buttons are not accessible on a small desktop screen.

New Known Issues

Ticket	Component	Description
TD-19150	Transformer Page	On Photon, `unnest` transform fails if `pluck=true`. **Workaround:** The `pluck` parameter forces the removal of the unnested values from the source. You may be able to use the `replace` transform on the source column to remove these values.
TD-19032	Transformer Page	Swapping rapidly between source datasets that have already been edited may cause a `No samples found` error. **Workaround:** Log out and log in again. Perform your dataset swap as needed.
TD-18933	Transformer Page	You cannot load a dataset that utilizes another dataset via join or union three levels deep. Example: three datasets (`Level1`, `Level2`, `Level3`) each integrate `ref_dataset` via join. You union `Level1` and `Level2`. Then, when you try to union those two into `Level3`, you get an error. **Workaround:** You can generate results for the lower-level datasets and then create a new wrangled dataset from these results. However, you no longer automatically inherit changes from the source dataset(s).
TD-18836	Transformer Page	`find` function accepts negative values for the start index. These values are consumed but produce unexpected results. **Workaround:** Use non-negative values as inputs.

TD-18746	Transformer Page	When Photon is enabled, previews in the data grid may take up to 30 seconds to dismiss.
		Workaround: This issue is related to the display of suggestion cards. Although it's not an ideal solution, you can experiment with disabling the display of preview cards in the data grid options menu. See *Data Grid Panel*.
TD-18538	Connectivity	Platform fails to start if Trifacta user for S3 access does not have the ListAllMyBuckets permission.
		Workaround: Please verify that this user has the appropriate permissions.
TD-18288	Installer/Upgrader	In Release 3.1.2 and earlier, any datasource that has never been used to create a dataset is no longer available after upgrade.
		Workaround: The assets remain untouched on the datastore where located. As long as the user has read permissions to the datastore area, the assets can be re-imported into the platform for Release 3.2 and later.
TD-18268	Transformer Page	If you profile a wide column (one that contains many characters of data in each cell value), the machine learning service can crash.
		Workaround: Restart the machine learning service. If visual profiling of the column is important, look to split the column into separate columns and then profile each one individually.
TD-18093	Transformer Page - Tools	Changes to a dataset that generates new columns can break any downstream lookups that use the dataset.
		Workaround: If the lookup breaks, you can recreate it.

TD-17713	Connectivity	Preview of Hive tables intermittently fails to show table data. When you click the Eye icon to preview Hive table data, you might see a spinner icon.
		Workaround: To workaround, preview data on another Hive table. Then, preview the data on the first table again. If you do not have another table to preview, try previewing the Hive table three times, which might work.
TD-17677	Administration/Configuration	Remove references to Zookeeper in the platform.
		Workaround: As of Release 3.2, the Trifacta platform no longer requires access to Zookeeper. However, removal of all references in the platform requires more work, which will be completed in a future release.
TD-17657	Transform Builder	`splitrows` transform allows splitting even if required parameter `on` is set to an empty value.
		Workaround: Make sure you specify a valid value for `on`.
TD-17333	Transformer Page	Sorting on a Datetime column places 00:00:00 values at the bottom of the list when operating on the Javascript running environment.
		Workaround: This issue does not appear in the Photon running environment or in jobs executed in Photon or Hadoop Pig. See *Configure Photon Running Environment* .
TD-16419	Transform Builder	Comparison functions added through Builder are changed to operators in recipe

TD-15858	Connectivity	Importing a directory of Avro files only imports the first file when the Photon running environment is enabled.
		Workaround: You can try re-exporting from the source system in a different format or importing the data when the JavaScript-based running environment is enabled. For more information on how to re-enable, see *Configure Photon Running Environment* .
TD-14622	Script Infrastructure	Python and Java UDFs accept inputs with zero parameters.
		Workaround: Insert a dummy parameter as part of the input.
TD-14131	Compilation/Execution	`splitrows` transform does not work after a backslash.
TD-12283	Installer/Upgrader/Utilities	Platform cannot execute jobs on Pig that are sourced from S3, if OpenJDK is installed.
		Workaround: Install Oracle JDK 1.8 before installing the Trifacta platform. See *System Requirements*.

TRIFACTA