

# Release notes for version 3.6.3

**Important!** Create a backup copy of your projects before updating to the new version. Projects saved in the new version can't be opened in earlier versions.

## Bugfixes

- Degraded performance on wide tables (>10'000 columns).
- Column name deduplication sometimes could change names of existing columns, or column names that didn't require deduplication.
- Win 10 Fall Creator's Update crashed application when progress popup is shown.
- Importing from Excel spreadsheets left temporary files undeleted.

# Release notes for version 3.6.2

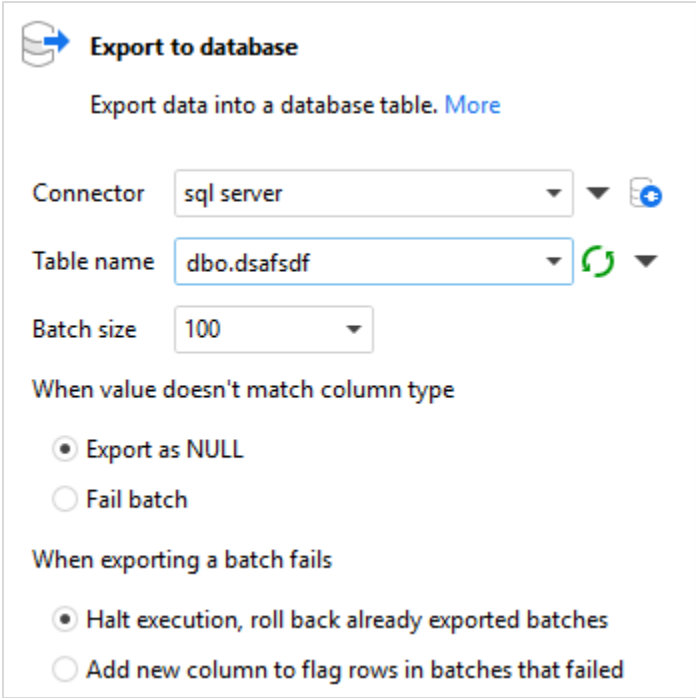
**Important!** Create a backup copy of your projects before updating to the new version. Projects saved in the new version can't be opened in earlier versions.

## Changes to existing transformations

### *New modes in "Export to database"*

Previously EasyMorph converted values with mismatching types (e.g. numbers in a text column) to NULL when exporting to a database. Now it has the option of failing in case of type mismatching.

Also a new mode has been added that appends a column with batch export status (Success or Failed). Even if 1 row in a batch fails to export, then all rows in the batch are flagged as failed. The status can later be used to split dataset rows into successfully exported and failed ones, and process them differently.



*Screenshot 1: New modes in "Export to database"*

## Multiple commands in "Run Program"

The "Run program" transformation now allows specifying multiple calls to the same application (or Windows shell commands) in one transformation. It also supports inserting project parameters into commands.

**Run program**  
Run an external application. [More](#)

Application  
C:\Program Files (x86)\7-Zip\7z.exe

Working directory (optional)

Arguments

Text (parameters supported e.g. {MyParam})

a {Archive} file1.txt  
a {Archive} file2.txt  
a {Archive} file3.txt

Calculated

Halt if exit code is not 0

Capture output

Screenshot 2: Multiple commands in "Run Program".

## Miscellaneous

- When a called project contains one table it's now assumed default, even if no default result table specified explicitly in the project
- Sandbox and Input now have the option to choose decimal separator when pasting data from the clipboard

# Release notes for version 3.6.1

**Important!** Create a backup copy of your projects before updating to the new version. Projects saved in the new version can't be opened in earlier versions.

## Breaking changes

Changes that can make EasyMorph produce different results with the same source data, or behave differently than previous versions.

### *New parser for delimited text*

Starting from this release EasyMorph employs a new parser for delimited text data. The new parser is fully compliant with RFC 4180 and supports quoted line breaks (the old parser didn't support it).

Old behavior	New behavior
The text below is processed as <b>2</b> lines. "AAA BBB", 123	The text below is processed as <b>1</b> line. "AAA BBB", 123

The change affects the following transformations:

- Import delimited text file
- Split delimited file
- Sandbox (when pasting tab-delimited data)
- Input (when pasting tab-delimited data)

To help detect text files that can be affected by the change, the installer of ver.3.6.1 includes a command line utility `CompareCsvParsing.exe` that detects discrepancies when a text file is parsed with the old parser (used in version 3.6 and below) and with the new one (used in version 3.6.1 and above). The utility is standalone and can be used independently from the EasyMorph executable. It accepts wildcards as argument and can be used to check multiple files at once.

### *Renamed keywords in the system function*

For the purpose of more consistent naming, two keywords used in the system function have been renamed: `programpath` renamed to `programdir`, and `currentpath` renamed to `currentdir`.

Old behavior	New behavior
<code>system("programpath")</code> <code>system("currentpath")</code>	<code>system("programdir")</code> <code>system("currentdir")</code>

## Upgraded requirement for .NET version

Versions 3.6.1 and above now require Microsoft .NET framework version 4.5.1 (or higher). Previous versions required .NET 4.5. Most probably, .NET 4.5.1 has already been installed on your computer with regular security updates for Windows. If for some reason it hasn't been installed, you can download it here: <https://www.microsoft.com/en-us/download/details.aspx?id=40773>

## Changes to existing transformations

### *“Export to Tableau” supports timestamps and integers*

Previously timestamps (i.e. dates with time part) were exported into Tableau as only dates, without the time part. Starting from this release, if a column contains at least 1 date with time and has type “Date” in the “Export to Tableau” transformation then its Tableau data type set to DATETIME and its values are exported as timestamps.

Similarly, numbers previously were always exported as floats. Starting from this release, if a column contains only integer numbers its Tableau data type is set to INTEGER and all its values are exported as integers. If the column has at least 1 non-integer value, or value that is outside of the integer range, then entire column is exported as floats.

Old behavior	New behavior
<b>1000</b> exported as <b>1.0E+3</b> (float).	<b>1000</b> exported as <b>1000</b> (in a column with integers only).
<b>1/1/2017 12:12pm</b> exported as <b>1/1/2017</b> (no time part).	<b>1/1/2017 12:12pm</b> exported as <b>1/1/2017 12:12pm</b> (time part preserved).

### *“Append” mode for “Export to delimited text”*

The “Export to delimited text” transformation now has a new mode that allows appending to an existing file, or creating it if it didn't exist. No column matching is done – columns are written in the same order as in the dataset, regardless of the column names in already existing data.

The “Append” mode can be used for logging, or dumping particular records, e.g. records that didn't pass data quality checks.

## What's new

### *Database connectors specified by parameter*

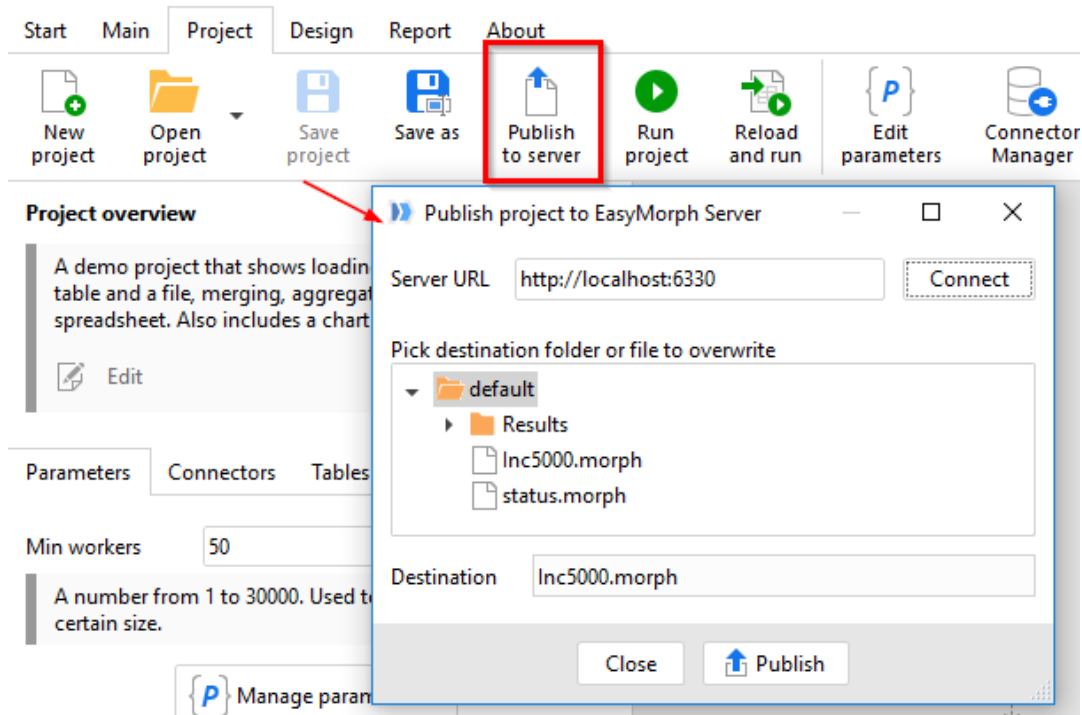
Three transformations now allow specifying a database connector using a parameter:

- Database command
- Export to database
- Delete matching database rows
- Select matching database rows

Parameters can be used for specifying names of shared and embedded connectors. Note that EasyMorph queries do not support specifying connector names by parameter yet (it will be added in v3.7). Therefore transformations that use a query (“Import from database” and “Delete database rows”) do not support specifying connector names by parameter too.

### *Publishing projects to EasyMorph Server*

All desktop versions of EasyMorph now can publish projects directly to EasyMorph Server v1.2 and up. New button “Publish to server” has been added to the Project menu.

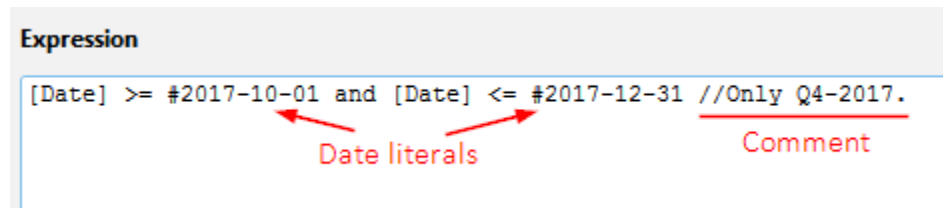


*Screenshot 3: Publishing to EasyMorph Server.*

## Comments and date literals in expressions

Now it is possible to insert comments in expressions. Start a comment with a double-slash (//). Everything between the double-slash and the end of line will be considered a comment and ignored during expression evaluation. A comment can't span across multiple rows.

Date literals start with the pound sign (#) and must have the ISO format only, i.e. #YYYY-MM-DD. Time is not supported in date literals. Note that there must be no space between # and year. See the screenshot below:



Screenshot 4: Date literals and comment in expression

## New system value 'projectdir'

The `system()` function has got new keyword 'projectdir' which makes the function return the path to the folder of current project, or nothing, if the project has never been saved. Example:

`system('projectdir')` returns 'D:\Projects'.

## Miscellaneous

- The "Import from delimited text" transformation now has option "Ignore quoting" in Advanced Settings. It makes all double quotes imported as regular characters.
- The hot key for adding a transformation to a table has changed from Ctrl+A to Ctrl+Q because the previous combination (Ctrl+A) conflicted with the default behavior in text boxes (select all text).
- Now, when searching for transformation in the list of available transformation both regular and start transformations are shown.
- Ctrl+click'ing a transformation now re-runs it, even if it's already calculated.

# Release notes for version 3.6

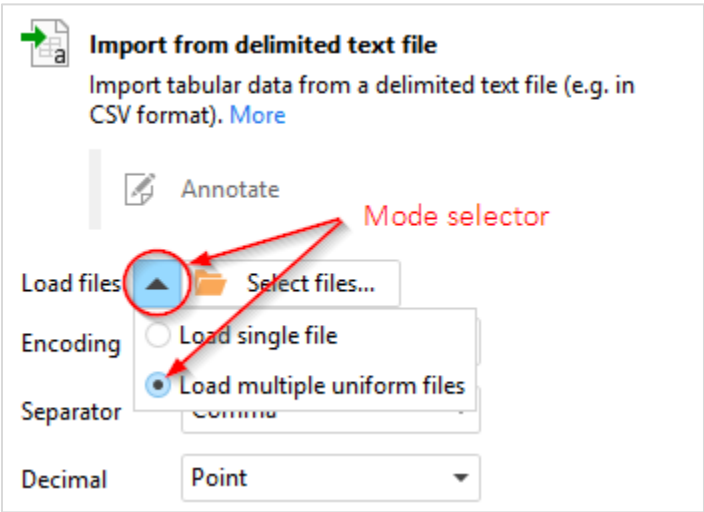
**Important!** Create a backup copy of your projects before updating to the new version. Projects saved in the new version can't be opened in earlier versions.

## What's new

### Multiple file loading

Previously, loading multiple uniform files (i.e. files with the same type and similar set of columns) required using iterations. Starting from this release, all file import transformations allow loading multiple uniform files at once without using iterations. Loaded files are automatically concatenated into one table.

To switch to loading multiple files click the black arrow as in the image below:




Screenshot 5: Load mode selector.

Files to load can be selected either explicitly, by selecting particular files in target folder, or by applying a filter which can be a search substring, a wild card, or a regular expression.

Note that the free edition has a limit of 30 uniform files loaded per project.

### New transformations

 **Fill Right** transformation is similar to the “Fill down” transformation. It fills empty cells to the right, in specified row. This transformation helps dealing with spreadsheets where table headers contain merged cells.





**Download File** transformation fetches a file over the internet using the HTTP(S) or the FTP network protocol. It supports basic login/password authorization.



**Import Fixed Width Text** transformation allows loading text files with columns of fixed width (e.g. logs). It has an interactive preview pane that allows visually splitting text into columns.

Date	Time	LineNo	Record Type	Message
06/11/2017	04:32:33 PM	000001	INFO	Job started manually.
06/11/2017	04:32:33 PM	000002	ERROR	Job failed: 1 error(s).
06/11/2017	04:32:33 PM	000003	ERROR	Scheduler: Error:Couldn't read shared data
06/11/2017	04:37:22 PM	000001	INFO	Job started manually.
06/11/2017	04:37:22 PM	000002	ERROR	Job failed: 1 error(s).
06/11/2017	04:37:22 PM	000003	ERROR	Scheduler: Error:Couldn't read shared data
06/12/2017	08:53:51 AM	000001	INFO	Job started manually.
06/12/2017	08:53:52 AM	000002	ERROR	Job failed: 1 error(s).
06/12/2017	08:53:52 AM	000003	ERROR	Scheduler: Error:Couldn't read shared data
06/12/2017	08:57:26 AM	000004	INFO	Job started manually.
06/12/2017	08:57:26 AM	000005	ERROR	Job failed: 1 error(s).
06/12/2017	08:57:26 AM	000006	ERROR	Scheduler: Error:Couldn't read shared data

Screenshot 6: The preview pane of the Import Fixed Width Text transformation.



**Split Fixed Width Text** transformation is similar to the “Import fixed width text” transformation, but splits an already loaded column (e.g. loaded from a database or spreadsheet).



**Split Delimited Text** transformation is similar to the “Import delimited text file” transformation (introduced long time ago), but splits an already loaded column into new columns. It can be used for instance for dealing with CSV data imported from a spreadsheet, or a database.



**Status** transformation is a workflow transformation. When reached it displays a custom message. Typically, it's used for signaling on project execution progress. For desktop versions of EasyMorph the message text appears in the “Run progress” dialog window. For EasyMorph Server the message appears as task status in the web-console, and also is logged into the task log. Custom statuses of called/iterated projects pop up into their calling projects, which allows displaying iteration progress (e.g. “Processed 15 files out of 132”).



**Regular Expression** transformation find matches for specified [regular expression](#). The transformation works in two modes: find 1<sup>st</sup> match, and find all matches. In the latter case, new

rows can be inserted into the table to store matches beyond 1<sup>st</sup>. Most common use cases for the transformation include data quality checks and text parsing.



**Select By Lookup** transformation select columns which names exist in another (lookup) table. It is intended for advances scenarios in which a resulting set of columns depends on user input or calculation logic.



**Convert Data Type** transformation converts text dates in one or more columns into number dates. The date format in selected columns can be auto-detected by pressing “Detect” button, if it matches one of 1,000+ formats known to EasyMorph. If no format detected then it should be entered manually in transformation properties. The format specification is the same as for the [format\(\)](#) function.

### *Changes to existing transformations*

**Export to Tableau** transformation now allows publishing data sources to Tableau Server.

The Concatenate function of **Aggregate** now allows specifying a separator and concatenating only distinct values.

**Running Total** transformation has got a [competition ranking](#) mode in addition to the default dense ranking.

**Select Matching Database Rows** and **Delete Matching Database Rows** transformations now allow matching on dates (without times).

**Iterate** transformation now runs faster and is more memory-efficient.

### *New functions*

**system(keyword)** – a special function that return various system values. The keyword must be a text constant. References to fields and expressions are not supported. Currently supported keywords:

Keyword	Description
<b>projectpath</b>	Full path to the current project including file name.
<b>projectfilename</b>	Current project’s file name only.
<b>programpath</b>	Full path to morph.exe.
<b>currentpath</b>	Working directory.
<b>user</b>	Current user. Replaces function user().
<b>callerpath</b>	Full path of the calling project including file name.
<b>utcoffset</b>	UTC offset to the system time zone ( e.g. "-04:00:00").

Example:

`system('projectfilepath')` returns 'C:\Documents\myproject.morph'.

**filename(text)** – extracts the filename from a full file path. Example

```
filename('C:\Documents\myproject.morph') returns 'myproject.morph'.
```

**combinepath(path1, path2)** – combines two paths into one, intelligently dealing with backslashes and relative paths. This function is convenient in cases when a file path is combined from a root folder (such as defined in an EasyMorph Server task) and a relative path. For example, when {Root} = 'C:\data':

```
combinepath({Root}, 'file.txt') returns 'C:\data\file.txt'.
```

```
combinepath({Root}, '..\file.txt') returns 'C:\file.txt'.
```

**div(number, number)** – returns the quotient of two numbers. Example:

```
div(10, 3) returns 3.
```

**rem(number, number)** – returns the remainder of quotient of two numbers. Example:

```
rem(10, 3) returns 1.
```

### *Custom HTML in generated documentation*

Auto-documentation now allows including custom HTML into the head and body of generated HTML page. This can be used for inserting corporate logos, links to other HTML pages, change logs, custom project metadata.

### *Miscellaneous*

- The visual condition builder in Query Editor now has undo/redo buttons.
- Added support for SAP HANA SQL dialect (for ODBC connections).
- Query preview now have a context menu invoked by right-clicking on datagrid cells.
- Expression editor has got a button for inserting date constants.
- Project tabs can be switched back and forth by pressing *Ctrl+Tab* and *Ctrl+Shift+Tab*.
- The “Input” transformation can now be enabled in Plus license as an add-on feature.