

Release notes for version 2.8.1

This release is a cumulative update for patches 2.8.0.x with a few new transformations and functions.

What's new

New transformations

Pivot transformation creates a matrix table using values of a column (e.g. Year) as column headers. Its action is opposite to Unpivot transformation. Number of new columns created by Pivot transformation is limited to 1000.

SQL Command transformation sends an arbitrary SQL command to a database connector. It doesn't receive back any result from the database. This transformation can be used right after exporting to a file to instruct a database to bulk load the file, or trigger a stored procedure. Read more on this in our web tutorial (<http://easymorph.com/learn.html>), chapter "Export to files and databases".

New functions

when(expression, result) – a shorthand equivalent of if(condition, result, empty()).

iferror(expression, result) – a shorthand equivalent of if(iserror(expression), result, expression).

compact(text) – removes repeating spaces inside a text value.

round(number, N) – round a number to N decimals.

Bugfixes

- Failed import of .xls files with one-letter sheet/range names.
- Failed import of .xls files with size ~15MB and particular structure.
- Failed import of .xlsx files programmatically generated by POI/NPOI libraries.
- Failed import of .sas7bdat with dates encoded with less than 8 bytes

Release notes for version 2.8

Important! Create a backup copy of all your projects before updating to the new version. This release modifies the project file structure and therefore projects created or saved in this version can't be opened in older versions of EasyMorph. You might also need to make edits in your existing EasyMorph projects in order to migrate them to the new version.

Behavior changes

1. Auto-naming has slightly changed and now it's less confusing. Although you might need to edit some transformation settings and expressions in existing projects in order to make them work in the new version. If you extensively used automatically generated names (which is not recommended as such names are not descriptive) then consider inserting **Rename** transformations to correct auto-generated names.

Old behavior	New behavior
When a transformation creates (or appends) a column with existing column name its name automatically gets suffix "(N)" starting from N=1 to keep field names unique. Example: <ul style="list-style-type: none"> • Amount • Amount(1) 	When a transformation creates (or appends) a column with existing column name its name automatically gets suffix "(N)" starting from N=2 to keep field names unique. Example: <ul style="list-style-type: none"> • Amount • Amount(2)

2. **Aggregate** now works much faster but returns unsorted results. Apply additional **Sort** transformation after **Aggregate**, if needed.

Old behavior	New behavior
Resulting table of Aggregate transformation is sorted by fields that were used for grouping.	Resulting table of Aggregate transformation is not sorted.

3. **Aggregate** now removes empty row if it's present.

Old behavior	New behavior
Resulting table of Aggregate transformation may include an empty row if source table contained empty lines.	Empty lines are always removed from a resulting table of Aggregate transformation.

4. **Fill Down** transformation now fills in zero-length text cells, not just empty (null) values.

Old behavior	New behavior
Fill Down fills in: <ul style="list-style-type: none"> • Empty cells 	Fill Down fills in: <ul style="list-style-type: none"> • Empty cells • Zero-length text cells

5. **Filter** transformation with no values selected now does nothing. This allows temporarily disabling filtering without removing the transformation.

Old behavior	New behavior
Filter with no values checked removes all rows.	Filter with no values checked <u>keeps</u> all rows.

6. Command-line options **/load** and **/run** are deprecated. Option **/document** is now intended for use only in the command line mode.

Old behavior	New behavior
Combination of options /c and /document generates documentation in the command line mode.	Combination of options /c and /document generates documentation and executes project in the command line mode.

7. We're switching to a new license key format that supports EasyMorph Plus edition. EasyMorph's technical support will send out updated keys to all customers that purchased a license.

Known issues

Noticeably longer run time of projects with large number (hundreds or thousands) of iterations.

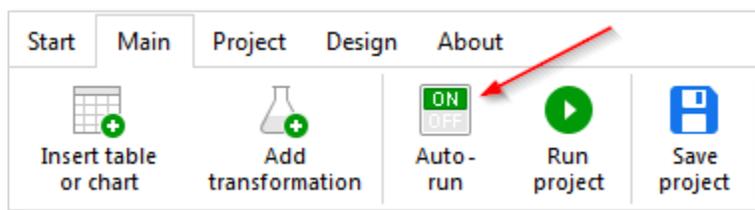
What's new

This release introduces a number of significant improvements and new features of the in-memory engine and user interface:

Pro-active background calculation (Auto-run mode)

EasyMorph now attempts to pro-actively calculate in the background all uncalculated transformations when any of the following happens:

- Properties of a transformation are changed
- A transformation is added, removed or moved
- A parameter is changed



This behavior is called Auto-run mode. It can be switched on and off from the main toolbar.

When Auto-run is off EasyMorph behaves as in previous versions – i.e. requires pressing *Run project* button to start calculations. When Auto-run is on, calculations are performed automatically. Tables and charts are updated in the background while you keep working with the project. To stop a running background calculation just turn Auto-run off.

In Auto-run mode EasyMorph silently stops on so called *side-effect transformations* – i.e. transformations that modify external files or trigger external applications. Currently, all export transformations as well as **Run Program** and **Iterate Program** are considered side-effect transformations. They have to be run manually (by pressing *Run project* or F5), even in Auto-run mode.

Auto-run state is saved with project. Auto-run mode is ignored when a project is executed from the command line.

Asynchronous execution

Previously, the in-memory engine of EasyMorph executed transformations one after another, always one transformation at a time. Now it automatically executes independent transformation chains asynchronously, meaning that if two (or more) transformations don't depend on each other then they most likely will run at the same time in parallel. If, for example, a project loads a text file and imports data from a database – these operations will typically be performed simultaneously.

If a workflow requires explicit synchronization of two independent transformation chains (e.g. one creates a file and the other triggers an external application that reads from the file), the **Synchronize** transformation can be used.

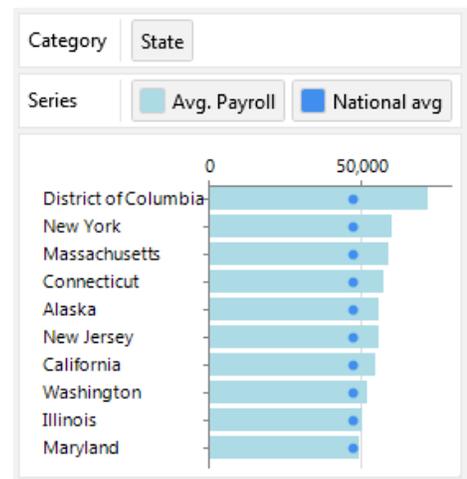
Asynchronous execution usually noticeably reduces project run times. No additional setup required for asynchronous execution – it works automatically out of the box, even for projects created in previous versions of EasyMorph.

Drag-and-drop charts

This release introduces a new type of model object – charts. Charts are derived from tables. When a chart is derived from a table, the table becomes the chart's data source, meaning that all changes to data in that table will be reflected in the chart. In Auto-run mode charts update automatically.

Features of charts:

- One category (dimension) with up to 50'000 category values.
- Multiple series (plots)
- 4 plot types (line, bar, dot, dash)
- One Y-axis
- Sort by category, series or a column in source table
- Scrolling (when enabled)
- Horizontal/vertical layout
- Auto-scaling



Categories and series can be added to a chart by simply dragging columns from its source table. Chart images can be copied into the clipboard in one click, and pasted in documents and presentations.

For EasyMorph editions with limited number of transformation one chart counts as one transformation.

New transformations

Synchronize transformation instructs EasyMorph to pause a transformation chain (not entire project), until specified table (which is another transformation chain) is calculated.

Either table is also a two-table transformation. It doesn't actually transform data but chooses which table (current or another one) is not empty and returns it. This transformation can be used for instance to organize conditional workflows, when calculation logic branches into two tables using derived tables and filters, where filters use a condition to make either one or the other table empty. Both branches can then be merged back using **Either table** transformation.

Wait – pauses transformations in a table for specified number of seconds. It can be used for workflows that for instance trigger an external process and then have to wait until the process finishes.

Reorder columns changes order of columns in a table.

Changes to existing transformations

The changes below don't affect calculations in projects created in older versions, although they are not backward compatible. Improved compression algorithm now makes some transformations run faster.

Aggregate transformation has got two new aggregation types – Concatenate and Any. Because of a new aggregation algorithm the transformation is now 30-50 times faster. See also the Behavior Changes section above.

Merge transformation is now 3-5 times faster.

Append has got 2 modes – *Append rows* (default mode, equivalent to previous versions), and *Append columns*. The latter mode is basically a “lazy join” for cases when you know ahead that both tables have the same number of rows and their order is correct. Also **Append** is now slightly faster.

Faster QVD reading/writing

The improved in-memory data compression algorithm is now better compatible with QVD format which allows EasyMorph to read and write QVD files faster (10-20MB/sec).

New data source: SAS files

Now EasyMorph supports importing .sas7bdat files. This feature is built using an unofficial open-source library and therefore it is not guaranteed to support all SAS files; however it has a good degree of compatibility and was tested on many files. Notice to SAS users: unlike SAS, EasyMorph loads data sets entirely in memory. Therefore for big datasets a significant amount of RAM might be required.

To load a SAS file use **Import from SAS** transformation, or simply drag the file into EasyMorph.

New data source: native connector for PostgreSQL databases

EasyMorph now includes a native connector for PostgreSQL databases. Native connectors don't require database client software (OLE DB) or ODBC drivers installed – they work out of the box after EasyMorph installation.

New functions

match(value, string, separator) – returns position of a value in a list of values which is a text string, delimited by separator. Example:

```
match(5, '5;55;555', ';') returns 1.
```

in(value, string, separator) – returns either true or false depending if a value is present in a list of values which is a text string, delimited by separator. Example:

```
in('A', 'a;aa;AAA', ';') returns FALSE.
```

pick(index, string, separator) – returns value from a list, delimited by a separator, by its index in the list. Example:

```
pick(3, 'Please don't stop the music', ' ') returns 'stop'.
```

user() – returns Windows user name.

New behavior of parameters

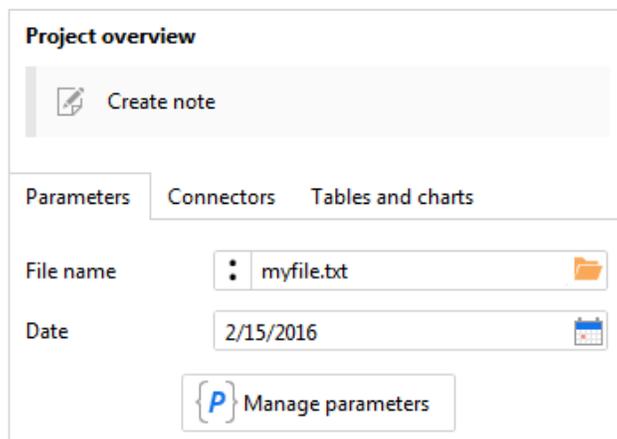
When a parameter changes, EasyMorph now discards results of all transformations that depend on this parameter, and, if Auto-run is on, recalculates them automatically using the new parameter value. Therefore, when a parameter changes EasyMorph automatically updates all tables and charts in the project accordingly.

Also starting from this release, calculated parameters can refer to other calculated parameters. If a cyclical dependence occurs an error message will be displayed.

New sidebar pane – Project overview

Starting from this release, accessing project parameters and embedded data connectors is possible through new sidebar pane *Project overview*. This pane is also used for creating a project annotation.

Shared data connectors are not shown in Project Overview as they are not embedded into project. They can be accessed as usually – in Connector Manager, invoked from menu Project, or by pressing F7.



New error notification workflow

As calculation errors can occur during background calculations (when Auto-run is on), a new error notification is introduced. Now the status bar shows clickable error counter when there are calculation errors. When clicked, it displays a list of errors. Clicking an error will open the transformation where the error occurred.

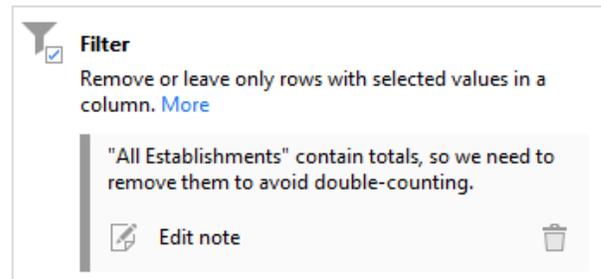


The error counter is also available when Auto-run is off.

Also, error messages related to a transformation are now displayed in the sidebar editor of this transformation.

Annotations

Now it's possible to create annotations for transformations, tables and a project. These annotations are also included in automatically generated documentation.



Improved run progress window

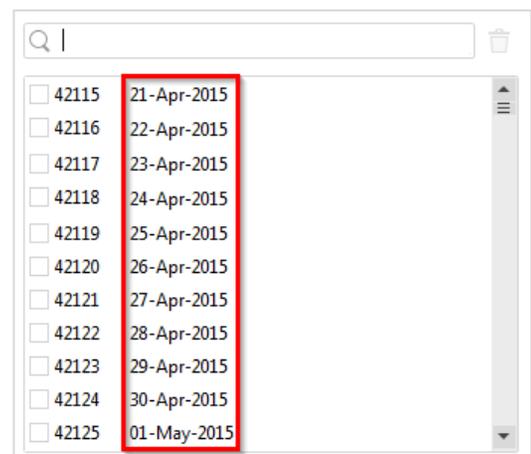
The run progress window now has **Stop** button and elapsed time counter. When **Stop** is pressed EasyMorph attempts to stop current transformation. Note that not all transformations can be stopped immediately – for instance all export transformations will keep executing until finished to avoid file corruption.



Improved viewing of column values

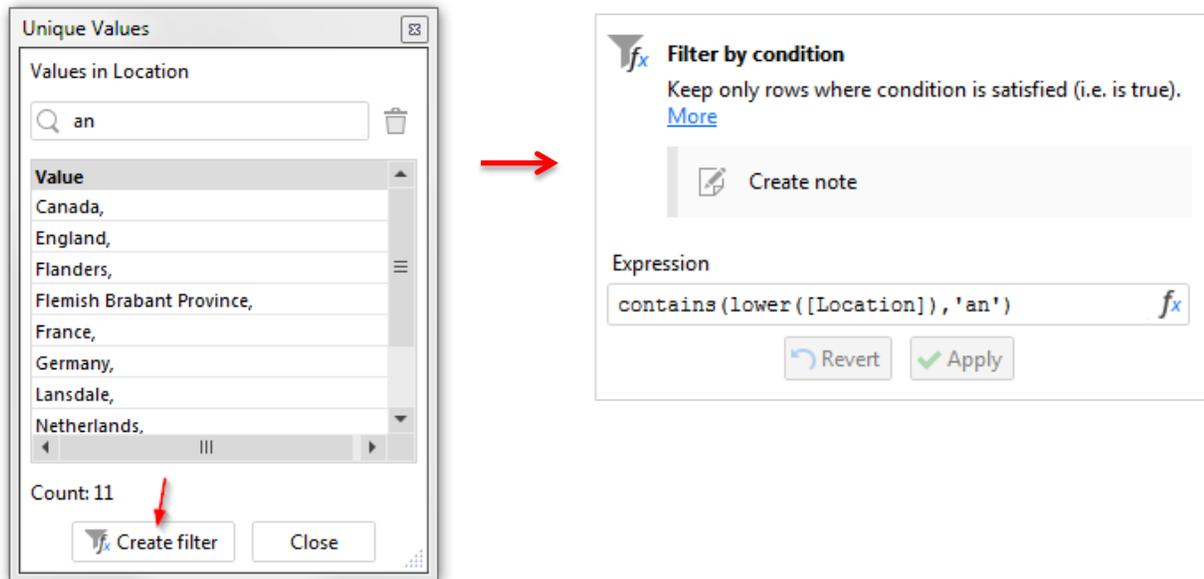
Dates are actually numbers in EasyMorph, even when they are shown as dates, according to column formatting. To simplify dealing with date-formatted columns, now values of such columns are displayed next to numbers in the Unique Values pop-up (invoked by double-clicking a column header), and in editors of **Filter** and **Group** transformations.

Also, when a column has many values (e.g. >100K) creating a filter for such column or viewing its values in previous versions could lead to a long delay that froze application. Now this delay has been greatly reduced and a non-freezing behavior implemented.



Create filter from Unique Values popup

Another improvement to the Unique Values pop-up is the ability to create new **Filter by expression** transformation in a single click. The transformation will filter column values to leave only values that contain the searched text string.



Miscellaneous UI improvements

- Toolbar ribbons Table and Transformation are now merged into one Design ribbon
- It is now possible to reorder columns by dragging
- It is now possible to remove columns and transformations by dragging them onto the workspace (the grey background area)
- Calculate button is removed from transformations' editors, and replaced with Remove button
- Transformation right-click menu now has two new commands:
 - *Calculate* – equivalent to Calculate button in older versions
 - *Discard result* – discards result of the transformation and all transformations that depend on it. Automatically triggers recalculation if Auto-run is on. This command can be used for instance to reload a single source file, without re-running entire project.
- Keyboard shortcut **Ctrl + B** sends a result of current transformation into a new Sandbox
- *Reload and Run* button has been moved from *Main* ribbon to *Project* ribbon.