

Release notes for version 3.0

Important! Create a backup copy of your projects before updating to the new version.

Behavior changes

- 1. Function **iferror(expression, expression)** behaved incorrectly and has been fixed in this version. Now it allows its first argument to evaluate to an error without error propagation. This function can now be used to handle calculation errors (e.g. before filtering by condition).

Old behavior	New behavior
iferror() always returned error when first argument is evaluated to an error. E.g. <code>iferror(1/0, "Infinity")</code> returned #Division by zero .	iferror() returns second argument when first argument is evaluated to an error. E.g. <code>iferror(1/0, "Infinity")</code> returns "Infinity".

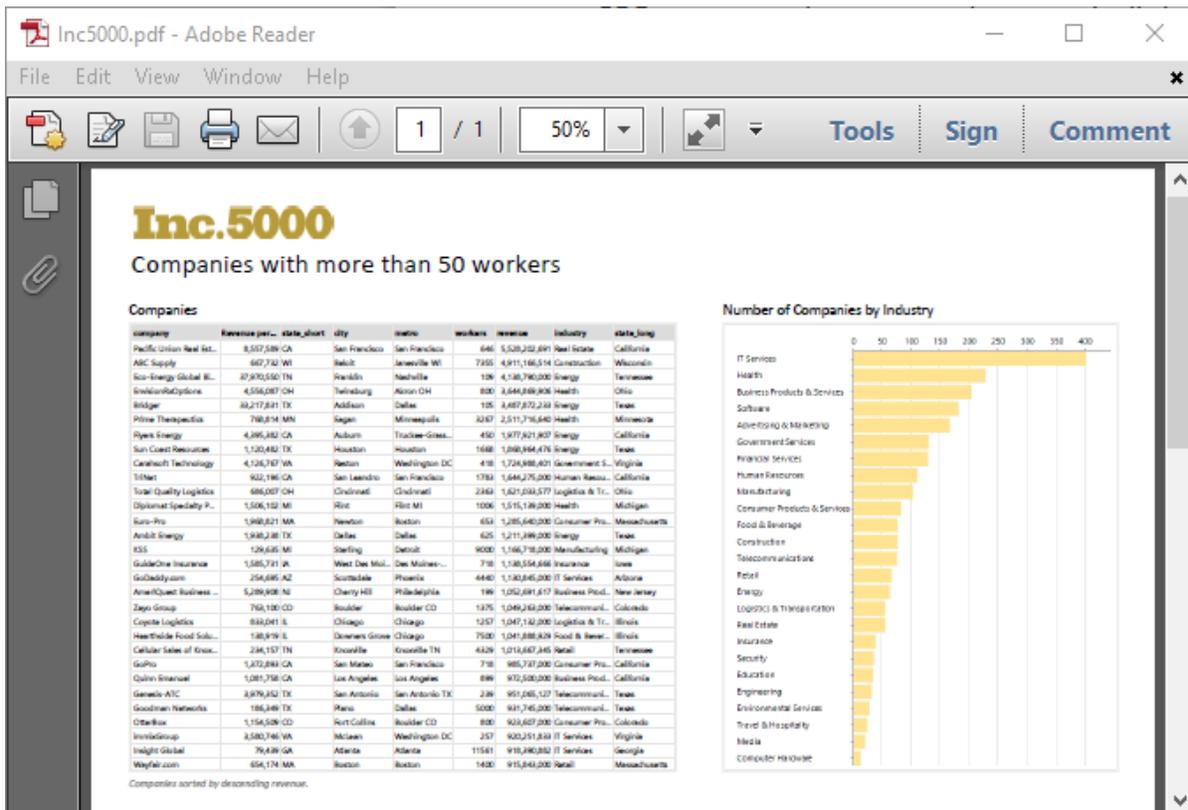
What's new

Basic PDF Reporting

PDF reporting is the headline feature of version 3.0. Now it's possible to create simple PDF reports in which one project tab corresponds to one report page. Selected charts and tables from a tab can be displayed on its report page with adjusted size and position. Tables can have totals. Report pages can also include custom titles (can be defined by expression), multi-line paragraphs of text, and images.

At this point PDF reporting has rather limited customization capabilities. The feature is intended for quick creation of simple reports. It's not positioned as a fully featured, pixel-perfect reporting tool. One notable limitation is that tables can't span across several pages.

PDF reports can be generated automatically by running projects from the command line with **/report** command (Professional edition only). Names of generated PDF files can be defined by expressions which can utilize project parameters (e.g. dates). Therefore it is possible, for instance, to iterate (with the help of "Iterate Program" transformation) across a list of countries, and create a separate report for each country.



Screenshot 1: PDF report generated by "Inc5000" example.

ODBC data sources in Connector Manager

Previously importing data from ODBC data sources was only possible by providing "Import from ODBC" transformation with a custom SQL query. Now ODBC is supported by Connector Manager (just like Oracle, or MySQL) which allows utilizing "Import from database" transformation with its drag-and-drop query builder and preview. The ODBC connector allows choosing a SQL dialect. Currently, supported SQL dialects are Oracle, MS SQL Server, MySQL, PostgreSQL, MySQL and SQL-2003. You can also use a compatible SQL dialect when applicable (e.g. PostgreSQL dialect can be used for ODBC connection to Amazon Redshift).

Note that "Import from ODBC" transformation will be deprecated in future versions of EasyMorph.

New transformations

 **Clean up** transformation removes particular values in a column by replacing them with empty cells. Its main use case is handling data quality issues.

 **Shift column** transformation shifts values of a column up or down, making it a new column. This transformation is intended for the cases when a reference to another row is required (e.g. comparisons with previous periods). It brings a value in another row of the same column into the same row as a new column so that it can later be used in expressions.



Call transformation runs another EasyMorph project once, then return its result table. “Call” transformation can pass its input dataset into the called project (it must use “Input” transformation in order to obtain it) therefore making possible sending a dataset from one project to another project directly in memory, without saving it to a file (as it was needed previously), which simplifies and speeds up inter-project data exchange. “Call” effectively makes possible creating custom transformations, where datasets can be transformed by external applications. The transformation is available in Professional edition only.



Input transformation obtains input dataset of “Call” transformation of a calling project. Output of this transformation is always empty and only populated during run time, when the project is called by another project. For development and testing purposes a test dataset can be pasted into “Input” transformation, just like into “Sandbox” transformation. “Input” transformation is available only in Professional edition.



Match transformation checks if values of a column match values in another table using one of the three modes: Contains, Starts With, Ends With. This transformation can be used, for instance, to detect and extract particular keywords from product descriptions or tweets.

Changes to existing transformations

The changes below don’t affect calculations in projects created in older versions, although they are not backward compatible.

File List and **Folder List** transformations now can recursively scan subfolders and retrieve lists of files and folders (respectively). The depth of scanning can be defined by a parameter. Note that you can scan folders by simply dragging them into EasyMorph.

Import from text file transformation now allows limiting number of loaded rows. This is convenient when developing transformations of large text files. The setting is available in advanced options. Note that advanced options now don’t include a setting to run external application before importing a text. This setting will keep working for existing projects that use it, although it will be deprecated in future versions. New projects should use “Run program” transformation.

Import from Excel transformation has got new option that allows converting too big or too small numbers with floating point into text, instead of generating an error.

New data source: XML files

EasyMorph now supports importing from XML files. You can choose whether to import entire XML file or only particular elements from it. Note that empty elements (e.g. <element></element>) are imported as empty text strings, not as empty cells. This makes it possible to distinguish cases when an element is empty and when it’s missing entirely. The latter case results in empty cells.

Known limitations: re-definition of namespaces in child elements and XDS are not supported.

New functions

substringcount(text, substring) – counts number of substrings encountered in the text. Example:

```
substringcount('one;two;three', ';') returns 2.
```

padstart(text, character, width) – pads the beginning of the text with the character. Example:

```
padstart('99', '0', 5) returns '00099'.
```

padend(text, character, width) – pads the end of the text with the character. Example:

```
padend('99', '0', 5) returns '99000'.
```

removetext(text, substring) – removes all occurrences of the substring from the text. Example:

```
removetext('Miss Mississippi', 'iss') returns 'M Mippi'.
```

removechars(text, characters) – removes particular characters from the text. Example:

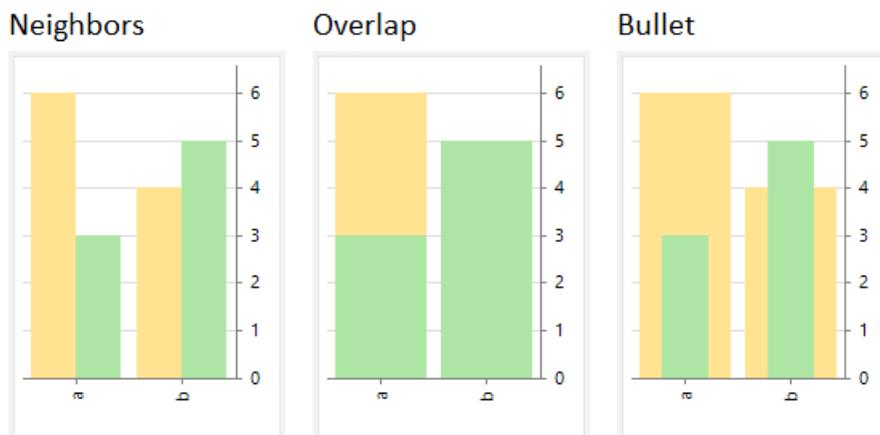
```
removechars('a1b2c3', 'abc') returns '123'.
```

keepchars(text, characters) – keeps only particular characters in the text, and removes all other. Example:

```
keepchars('a1b2c3', 'abc') returns 'abc'.
```

Bar composition in charts

Starting from this release bars in charts can be composed using one of the three modes:



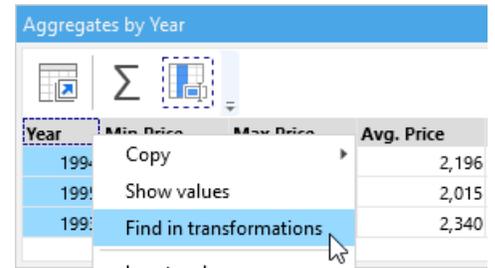
Screenshot 2: Bar composition types.

The Overlap mode can be used for designing stacked bar charts, or for showing positive and negative numbers in different colors. Bar composition has no effect in charts without bars.

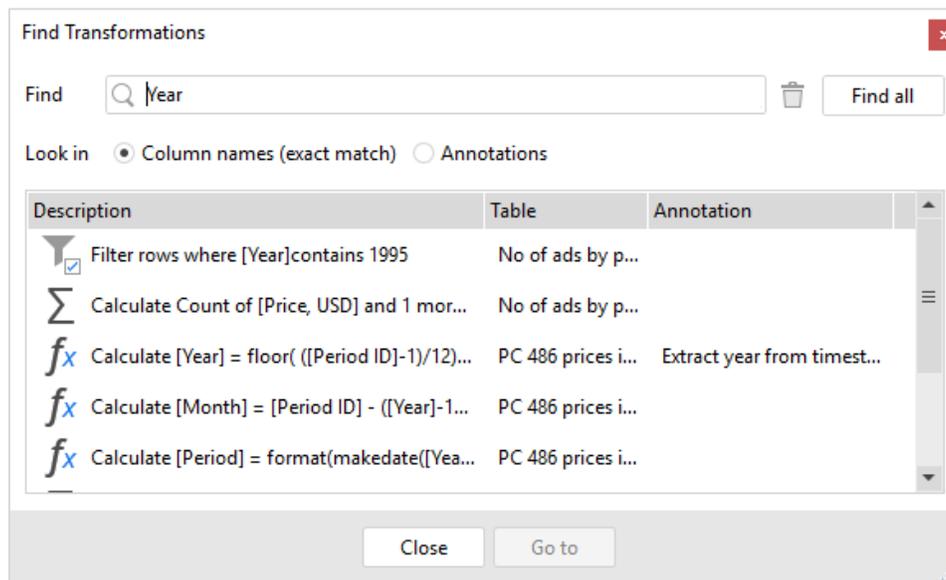
Find Transformations dialog

Version 3.0 introduces a long awaited feature – the ability to find all transformations that operate with certain column. This sometimes is needed when working with large projects that consist of hundreds of transformations in order to find a transformation that calculated or modified particular column.

Starting from this release it can be done using the Find Transformation dialog. To initiate a search – right click a column and choose “Find in transformations”. Alternatively, click “Find transformations” button on the toolbar and type in the column name (case-sensitive).



Year	Min. Price	Max. Price	Avg. Price
1994	Copy		2,196
1995	Show values		2,015
1996	Find in transformations		2,340

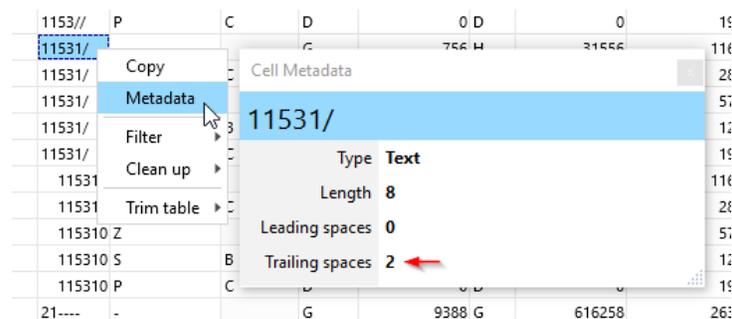


Screenshot 3: Find Transformations dialog.

This feature is especially helpful when key transformations are annotated, as it allows easily spotting them in the list. Annotations are searchable too.

Cell Metadata popup

Cell Metadata popup is a tool to quickly inspect a cell value – its type and some additional metadata (when applicable). Invoked by right-clicking a cell.



Screenshot 4: Cell metadata popup.

You can invoke the popup once, and then inspect cells as long as it stays open.

Miscellaneous UI improvements

- Button  next to file names to open the files using default programs associated with extension. Now it's possible, for instance, to open spreadsheets in Excel right from "Import from Excel" transformation sidebar editor.
- Reduced UI lags in projects with many columns (hundreds), and in projects with many charts.
- Long operations (e.g. opening a database connection) now display a popup with a progress bar.
- Displaying input dataset (when possible), when output dataset is not available.
- Snap-to-grid mode when objects are moved with depressed Shift key (available in both Report and Design modes).
- Calculation errors in cells are now shown in red.
- Support for pasting data without column headers (in Sandbox transformation).
- In the command line mode run errors are now shown in the terminal window as well.
- Similarly to the table data grid, the list of unique values of column a now uses text alignment to indicate value type (left for text, right for numbers).
- Redesigned Preview button in some import transformations (Import from text, Import from DB).
- Hitting horizontal scroll buttons leads to scrolling by a larger delta.
- New EasyMorph logo and related icons.

Updates to data drivers

- Updated Tableau drivers
- 10%-30% faster import from SAS files
- Added support for .xlsm files

Known Issues

Arabic, Hebrew and Asian languages are not supported in PDF reports.