

# Release notes v4.1

**Important!** Create a backup copy of your projects and connector repositories before updating to the new version. Projects saved in version 4.1 can't be opened in earlier versions.

## *A few words from the creators of EasyMorph*

In this version we're adding actions and functionality that we didn't introduce in the previous version (4.0) because we didn't want to postpone its release. This includes, first of all, actions for parsing/constructing JSON which are crucial for using web requests introduced in the previous version. Besides that we're beefing up analytical capabilities and started adding actions for advanced analytics and complex calculations.

On the Server side, we're steadily improving Server usability. This time we're adding Pages which can display custom HTML pages next to tasks and files, and a few new file/folder operations.

## Breaking changes

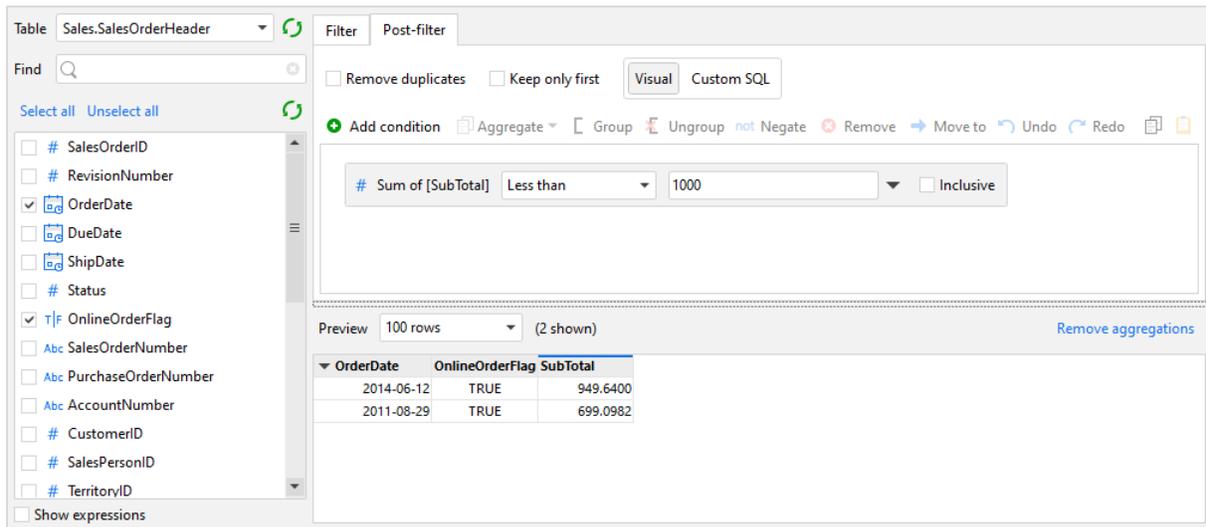
What changed	What it can break
When importing from a database, date 0001-01-01 was imported as 0.  Now it produces an error.	Projects that imported database tables where the zeroth date could occur now will fail with error.
The "Iterate" action returned empty dataset if the input dataset was empty or had no rows so the action didn't perform iterations. It was inconsistent with cases when the input dataset had 1 or more rows, because in this case the action adds columns with iteration parameter values.  Now the action returns a zero-length dataset with added columns which names are the names of iteration parameters.	Projects that expect that applying "Iterate" over an empty dataset returns an empty dataset (i.e. without columns).

## What's new

### *Post-filter in visual queries*

This release introduces post-filtering conditions which is a logical step after the previous release introduced aggregations. The conditions can filter the aggregated query result.

Post-filtering conditions under the hood are implemented using the SQL HAVING construct. The conditions are similar to regular filtering conditions with the exception that they only work with aggregated fields.



Screenshot 1: Post-filter in queries.

## New actions



The **Parse JSON** action parses a JSON contained in a cell into a tabular dataset. When applied to a column with JSONs, it automatically appends all parsed results into one tabular dataset. This action can be used after “Web Request” or “Iterate Web Request” actions in order to parse the response body.



The **Construct JSON** action allows creating nested JSON objects from single-row or multi-row datasets. Complex JSON objects should be constructed by applying the “Construct JSON” action multiple times in order to construct child elements first. Finally, the children elements are appended into one table and then packed into a final JSON object by using “Construct JSON” again. This approach allows creating nested JSON objects of any complexity.



The **Iterate columns** action is a highly advanced action for column-by-column processing of datasets. It passes columns of the input dataset one-by-one into the called module/project and appends the 1<sup>st</sup> columns of the resulting datasets back into one table. The actions helps process matrix datasets, arrange complex aggregations, and perform column filtering on condition.



The **Create buckets** action aggregates values into buckets (ranges) of fixed width. The action can be used to analyze distributions and prepare data for charting.



The **Make date/time columns** action provides a fast and convenient way to extract various date/time attributes from dates without using expressions. E.g. quarter numbers, ISO weeks, week days, etc.

 The **Linear regression** action creates a linear regression, calculates its coefficients and interpolates/extrapolates missing values, if necessary. The action can be used for analyzing trends and simple metric forecasting.



The **Statistical aggregation** action calculates a number of statistical functions:

- Median
- Percentile
- Standard deviation
- Standard error
- Variance

### *Changes in existing actions*

The **Import from SAS** and **Import from QVD** actions now allow importing only specific columns.

The **Export dataset** and **Export to QVD** actions now work much faster.

The **Import from delimited text** action now has an option for adding columns automatically if the number of values in a row exceeds the number of column headers.

### *New functions*

**code(text)** – Obtains the UTF-8 code (in the range from 0 to 2047) for the 1<sup>st</sup> character of the given text. The function is similar to function CODE in Excel. Example:

`code('ABC')` returns 65.

**iseven(number)** – Returns TRUE if the number is even, otherwise FALSE. Example:

`iseven(145)` returns FALSE.

**isodd(number)** – Returns TRUE if the number is odd, otherwise FALSE. Example:

`iseven(145)` returns TRUE.

**truncate(number)** – Returns the integral part of the number. Example:

`truncate(3.1415)` returns 3.

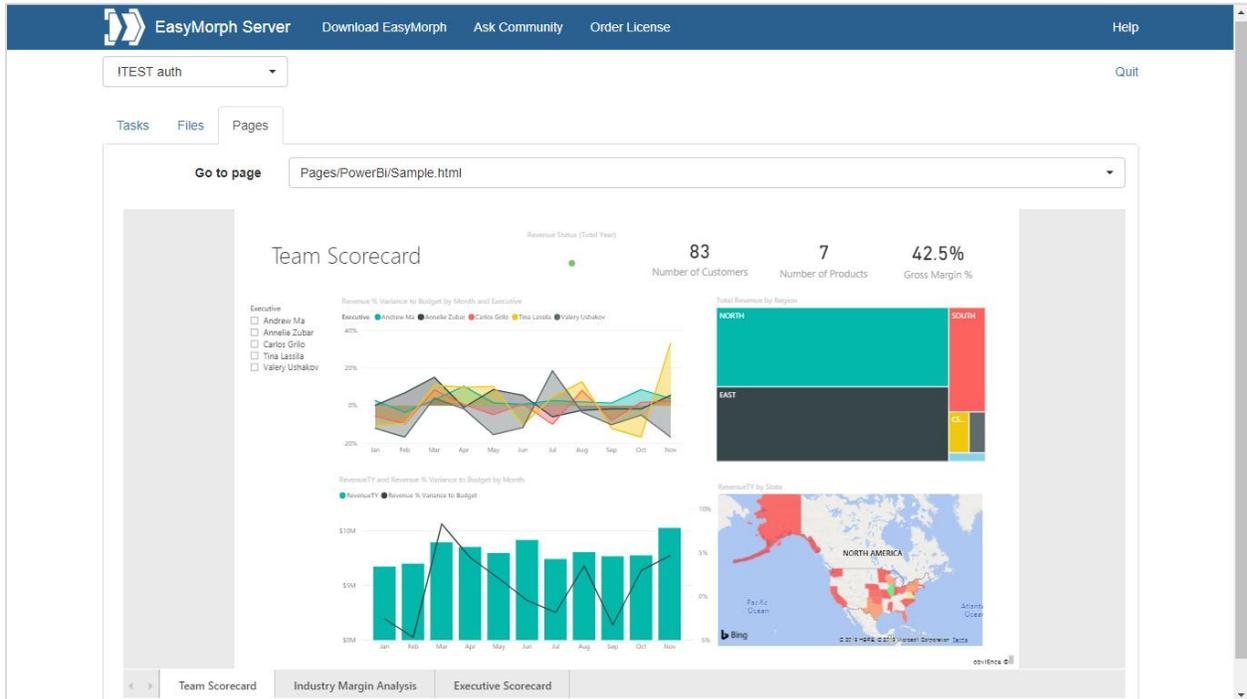
## What's new in Server

### *Pages*

Users of EasyMorph Server now can see custom HTML pages in a new tab called Pages (disabled by default). Such HTML pages can display various helpful information. For instance:

- Welcome page
- Quick tutorial with a few embedded YouTube videos.
- Embedded Power BI / Tableau Online / Google Data Studio dashboard, or a mix of them
- Charts showing EasyMorph Server usage statistics (RAM, CPU, etc.)
- An HTML table with result data generated by an EasyMorph task

In the screenshot below, the Pages tab contains a custom HTML page with embedded Power BI dashboard.



Screenshot 2: Custom HTML page in the Pages tab.

### Additional file/folder operations

- Create folder
- Rename folder
- Delete folder (must be empty)
- Rename file

### Miscellaneous

- New “Dismiss” button in the task log that dismisses error messages and changes task state from “Failed” to regular.

### Previous release notes

Link: [Release notes for v4.0.1.](#)