Release notes v4.4.1

A few words from the creators of EasyMorph

This release is mostly intended for cumulative bugfixes. We also used the opportunity to add a few minor features before the next major version is released.

What's new

Changes in existing actions

A new command "Run flow asynchronously" has been added to the **Tableau Command** action. The command allows triggering Prep flows published on Tableau Server. Note that the action only triggers a flow and doesn't wait until the flow finishes or captures any flow errors.

New functions

isemail(text) – returns TRUE if the provided text value has the form of an email address. The function simplifies validation of email addresses. Example:

`isemail('username@example.com')` returns TRUE.

hexencode(encoding, text) – converts the provided text value into a hex string. The text is assumed to have the specified encoding. Supported encodings:

- base64url
- base64
- utf-8
- urf-16

Example:

`hexencode('base64url', 'VmFsaGFsbGE=')` returns "56616c68616c6c61".

`hexencode('utf-8', 'Valhalla')` also returns "56616c68616c6c61".

hexdecode(encoding, hex_string) – converts the provided hex string into a text value. The text is assumed to have the specified encoding. Supported encodings are the same as for hexencode() described above. Example:

`hexdecode('base64url', '56616c68616c6c61')` returns "VmFsaGFsbGE=".

`hexdecode('utf-8', '56616c68616c6c61')` returns "Valhalla".

The two new functions allow operating with binary arrays using the hex representation. This can be convenient in cryptographic calculations, such as generating JWT tokens.

New email connectors

Two new email connectors have been added: “Amazon SES” and “Outlook.com”. The connectors offer a simpler way to configure connections to the respective email services. Note that “Amazon SES” only allows sending emails, but not receiving them.

New SQL dialect

We’ve added support for the “DB2 for AS/400” SQL dialect. It can only be used in an ODBC connector. The SQL dialect supports importing as well as exporting/updating/deleting data in DB2 for IBM i Series.

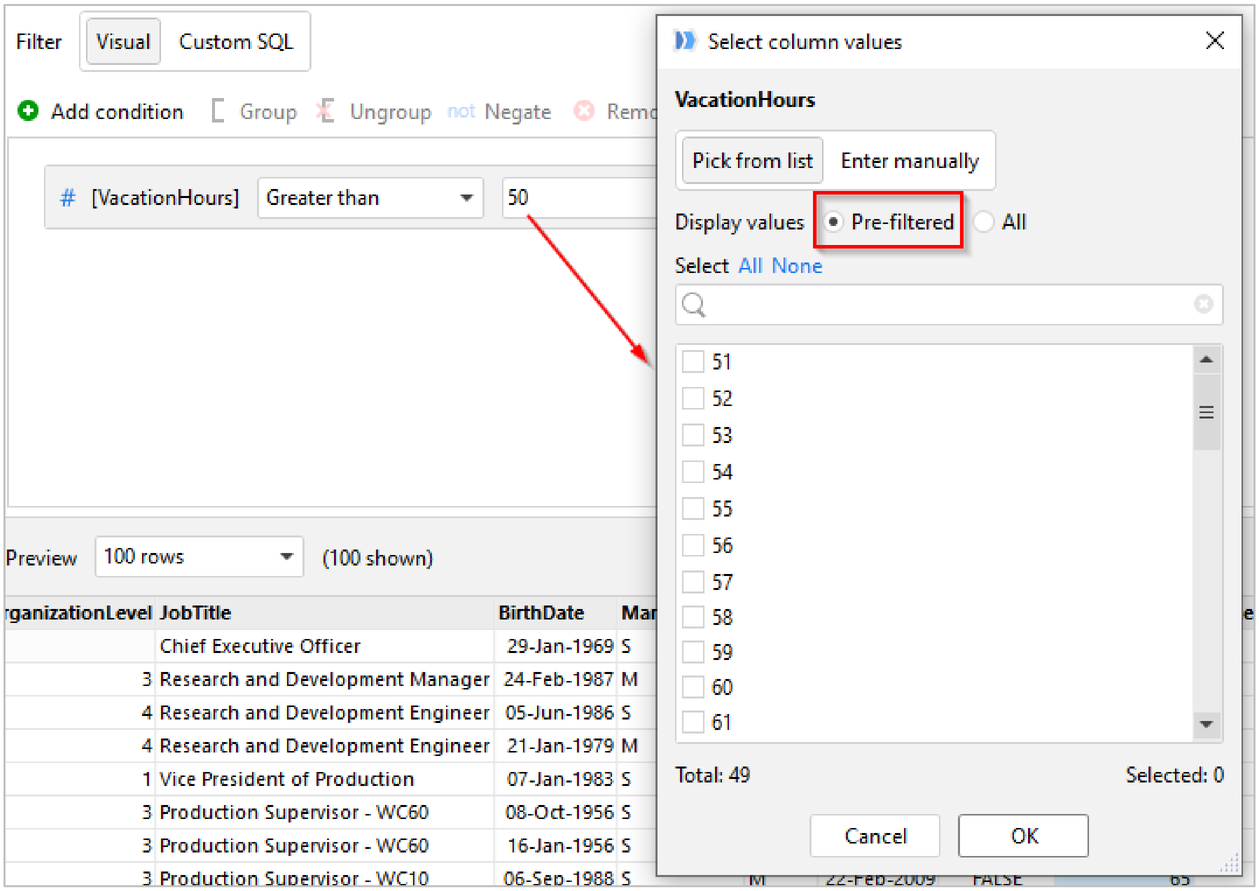
What’s new in Desktop

Query Editor: Pre-filtering in column values lists

When filtering values in a column, it may be convenient to pick from only the values that exist in the current query result, not from absolutely all unique values in the column. In the Query Editor it’s now possible to indicate that only pre-filtered values should be shown in a list of column values.

In the screenshot below, the “Select column values” dialog only shows values greater than 50, because the query filtering condition filtered out rows with [VacationHours] fewer than 50.

In some cases, pre-filtering may also speed up obtaining a list of column values.



# [VacationHours] Greater than 50

VacationHours

Display values ● Pre-filtered ○ All

☐ 51
☐ 52
☐ 53
☐ 54
☐ 55
☐ 56
☐ 57
☐ 58
☐ 59
☐ 60
☐ 61

Total: 49 Selected: 0

Preview 100 rows (100 shown)

rganizationLevel	JobTitle	BirthDate	Mar
	Chief Executive Officer	29-Jan-1969	S
3	Research and Development Manager	24-Feb-1987	M
4	Research and Development Engineer	05-Jun-1986	S
4	Research and Development Engineer	21-Jan-1979	M
1	Vice President of Production	07-Jan-1983	S
3	Production Supervisor - WC60	08-Oct-1956	S
3	Production Supervisor - WC60	16-Jan-1956	S
3	Production Supervisor - WC10	06-Sep-1988	S

Screenshot 1: Pre-filtering column values in Query Editor.

Miscellaneous

- New keyboard shortcut *Ctrl+Shift+D* creates a derived table in a newly created tab (group).
- New keyboard shortcut *Ctrl+Shift+M* moves the selected table to a newly created tab (group).
- The "Setup license" dialog now has a new look. The "Reset license" link has been removed from the Start page.
- The "Extend license" link is now always shown on the Start page when a valid license is applied.
- The "Main" toolbar now shows the "Buy license" button when no valid license applied. Pressing the button opens the in-app license purchase dialog.

Release notes v4.4

Important! Create a backup copy of your projects and connector repositories before updating to the new version. Projects saved in version 4.4 can't be opened in earlier versions.

Important! The new version introduces a number of BREAKING CHANGES. Do not roll out the new version blindly, without examining whether your current projects will be affected by the changes.

A few words from the creators of EasyMorph

This release is dedicated to improving EasyMorph fundamentals such as expressions, workflow composition, and ETL capabilities. It introduces (in experimental mode) new expression syntax, a mechanism of warnings, the "Skip" action as a simpler way to create conditional IF/THEN branching, and finally, an action for updating database tables – the "Update database table" action.

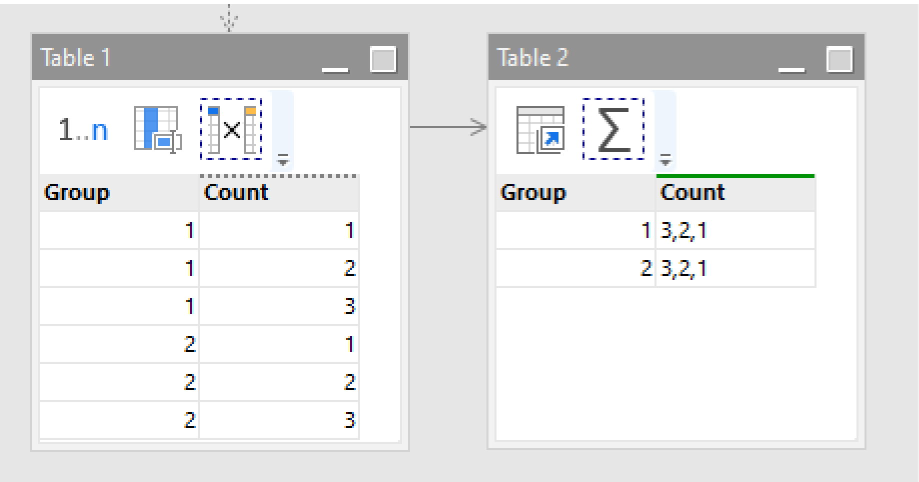
Another major novelty is the in-app license purchasing. Until now, we at EasyMorph processed all orders for new licenses and renewals manually. As the number of organizations that use EasyMorph has recently reached 300, manual license administration started showing its inefficiency. With this release, single-user licenses can be purchased and renewed automatically, right from the applications. Just click "Buy/extend license" to open the purchase wizard, provide necessary inputs, submit your order, and pay an online invoice. A new/renewed license key together and a PDF invoice will be emailed to you automatically. The key can also be installed automatically with a single click right in the purchase wizard. The whole process shouldn't take longer than 2-3 minutes.

Finally, we keep speeding up EasyMorph. The new version does aggregations and pivoting 2-3 times faster. Together with the new, faster expression evaluation mechanism, the improvements can speed up workflows by 20% on average and in some cases even more.

Breaking changes

Aggregate in the Concatenate mode

In previous versions the "Aggregate" action in the "Concatenate" mode concatenated values differently depending on whether grouping was applied or not. Without grouping, values were concatenated in the straight order. With grouping, values were concatenated in the reverse order.

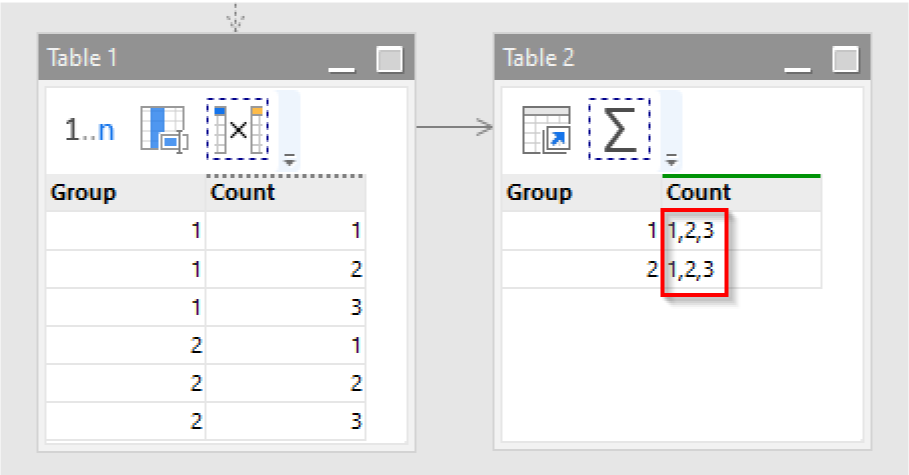


Group	Count
1	1
1	2
1	3
2	1
2	2
2	3

Group	Count
1	3,2,1
2	3,2,1

Screenshot 2: Concatenation order before v4.4.

Such behavior was inconsistent and sometimes required additional sorting prior to aggregation. The new version changes the concatenation order to the straight one in all cases to make the action's behavior consistent. However, the change is breaking. If a project was dependent on concatenation order, then it needs to be reviewed and edited in order to work correctly in version 4.4.



Group	Count
1	1
1	2
1	3
2	1
2	2
2	3

Group	Count
1	1,2,3
2	1,2,3

Screenshot 3: Concatenation order in v4.4

To simplify migration to the new version and finding projects that may require editing in order to keep working correctly in v4.4, we've created a project that scans EasyMorph projects in a given directory and finds those that can be affected by the change. For more detail see this forum topic: "Breaking change in v4.4: Aggregate with Concatenate and grouping".

Function "Hash()" now ignores trailing decimal zeros

In previous versions the **Hash()** function would return different values for two equal numbers if they have different numbers of trailing decimal zeros. For instance, Hash(1) returned "23I4Ee7FxMHOTZ910MntFw" while Hash(1.00000) returned "lwHMIA7K6qiJWDeRTKfw7w".

This behavior was inconsistent and could lead to wrong results. It has now been fixed. Now Hash(1.00000) returns exactly the same value as Hash(1) which is “23I4Ee7FxMHOTZ910MntFw”.

“Parse JSON” could insert extra rows by mistake

The “Parse JSON” action behaved incorrectly in some cases and could insert extra rows if the name of an XML element started with the name of another XML element (e.g. “FooBar” and “Foo”). This behavior has been fixed.

Sheet/range name length limits

The “Export to Excel” action previously allowed sheet names with more than 31 character and range names with more than 255 characters which could lead to creating workbooks that can’t be opened in Excel.

The new version enforces the restrictions on sheet and range names which are now not allowed to be longer than 31 and 255 characters respectively.

No locale-dependent formatting in SQL statements

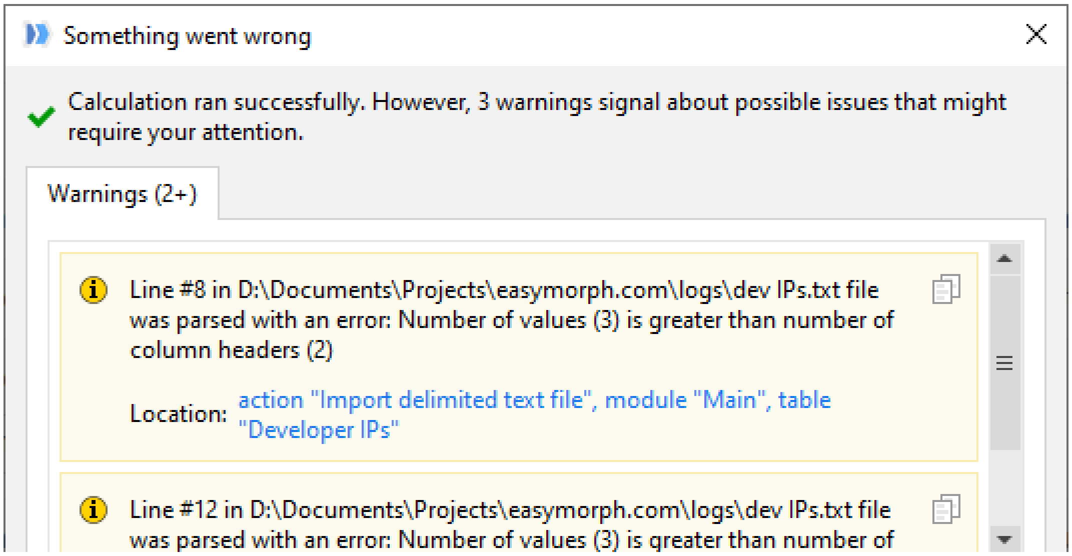
Previously, if a fractional number is inserted in a SQL query via parameter, it could be inserted with a comma as the decimal separator in certain Windows locales. Now this behavior is locale-independent and non-integer numbers are inserted with a dot as the decimal separator.

Also, in Custom SQL and Vertica SQL dialects dates were inserted in the format corresponding to the current Windows locale. Now they are inserted in the format corresponding to the SQL dialect.

What’s new

Warnings

To signal about suspicious issues that arise during project calculation, EasyMorph now generates warnings. Unlike errors, warnings don’t make projects fail because the detected issues aren’t deemed critical. Nevertheless, they may signal about possible data quality issues or errors in workflows.


Something went wrong

Calculation ran successfully. However, 3 warnings signal about possible issues that might require your attention.

Warnings (2+)

Line #8 in D:\Documents\Projects\easymorph.com\logs\dev IPs.txt file was parsed with an error: Number of values (3) is greater than number of column headers (2)

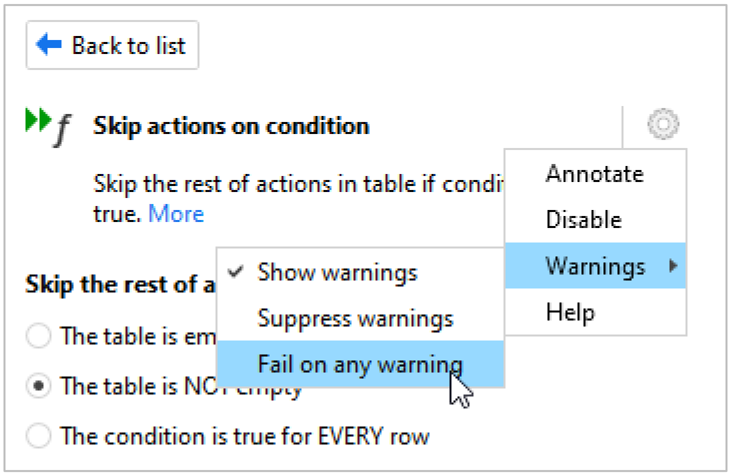
Location: action "Import delimited text file", module "Main", table "Developer IPs"

Line #12 in D:\Documents\Projects\easymorph.com\logs\dev IPs.txt file was parsed with an error: Number of values (3) is greater than number of

Screenshot 4: Warnings in the execution summary.

While warnings don’t cause a project fail, we recommend fixing the issues that trigger warnings and making sure your workflows are warning-free.

In the action settings, it is possible to make action fail on warning. In this case, a warning in that action will become an error and will force project to fail.



Screenshot 5: Warning options in action settings.

New expression syntax

The syntax of expressions in EasyMorph has been expanded with new features (see also [this post]()):

1) The unary minus operator. So instead of writing:

```
addmonths([Date], 0 - 1)
```

it is now possible to write:

```
addmonths([Date], -1)
```

2) Local constants. Constants must be assigned using the “let” operator before they can be used in expression. Syntax:

```
//Constant assignment.
let a = ([A] + [B]) / [C]
//Use of the constant in an expression.
([D] + a) / ([E] - a)
```

Notice that constant names don’t have to be wrapped in brackets, unlike field or parameter names. Constants help make expressions more readable. They also eliminate the need to create temporary columns in some cases.

3) The IF/THEN/ELSE operator.
Since the very early versions EasyMorph has been offering the **if()** function which works similarly to the IF() function in Excel.

Now, for better readability of expressions we’re adding the **IF/THEN/ELSE** operator, which is well known to Tableau users. Example:

```
IF isempty([FX Rate]) THEN error("Missing FX rate!") ELSE
[Amount] * [FX Rate]
```

The ELSE statement is optional. If omitted it returns an empty value:

```
//If FX rate is empty the result will be empty too.
IF not isempty([FX Rate]) THEN [Amount] * [FX Rate]
```

The new expression syntax is off by default. Currently, it has to be explicitly turned on in the Expression Editor. This is a temporary measure to make transitioning to the new syntax less risky. The new syntax will become permanent in one of the future versions. It is backwards compatible with the old syntax therefore you can safely enable it on in projects created in earlier versions.

Note that when the new syntax is enabled, the syntax error messages change too. If the workflow of one of your projects depends on analyzing error messages, then it will have to be adjusted to the new error messages.

Also, with the new syntax on, expressions are evaluated 2-3 times faster, especially on large datasets. On smaller datasets the speedup is insignificant.

New actions

The **Skip actions on condition** action makes EasyMorph skip the rest of actions in table (i.e. the actions after “Skip” action) if the specified condition is not fulfilled. Read more.

The **Update database table** action allows updating a database table with values from EasyMorph. The matching key fields must have only unique combinations of values in EasyMorph in order to avoid ambiguity. Read more.

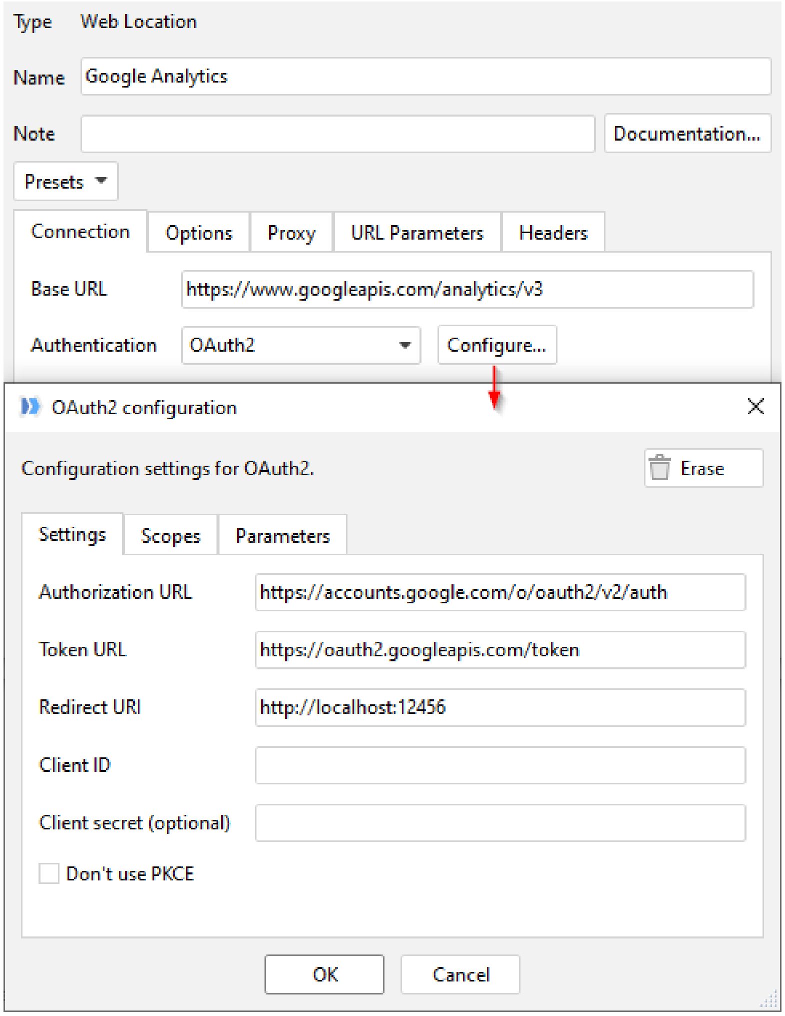
Changes to existing actions

The **Remove columns** and **Select columns** actions are now combined into one action named **Keep/remove columns**.

The **Import from delimited text** action now has an option (in the advanced settings) to treat numeric values as text in all cases, or only when they are quoted in the source file. With this option it's now possible to import all values as text, if necessary. Also the action is now 20-30% faster.

Support for OAuth in "Web location" connector

The "Web Location" connector now supports the OAuth2 authentication protocol. The enables accessing many web APIs (e.g. Google Analytics, or Salesforce) that require OAuth2 for authentication.



Type Web Location
Name Google Analytics
Note
Base URL https://www.googleapis.com/analytics/v3
Authentication OAuth2

OAuth2 configuration
Configuration settings for OAuth2.
Authorization URL https://accounts.google.com/o/oauth2/v2/auth
Token URL https://oauth2.googleapis.com/token
Redirect URI http://localhost:12456
Client ID
Client secret (optional)
☐ Don't use PKCE

Screenshot 6: OAuth in the "Web location" connector.

Note that OAuth1 is not supported.

New functions

columnexists(text) – returns TRUE if a column with provided name exists in the input dataset of the current action. Example:

`columnexists('Count')` returns TRUE is the current dataset contains column [Count].

Together with the "Skip actions on condition" the function can be used for optional calculations on columns that may or may not exist.

metadata(text) – returns various workflow metadata:

`metadata('current table')` returns the name of the table that contains the current action.

`metadata('current block')` returns the name of the tab that contains the current action. Note: in the next release, tabs will be renamed to blocks, hence the term.

`metadata('current module')` returns the name of the module that contains the current action.

isinteger(value) – returns TRUE if the argument is an integer number, and FALSE in all other cases. Examples:

`isinteger(123)` returns TRUE.

`isinteger('123')` returns FALSE.

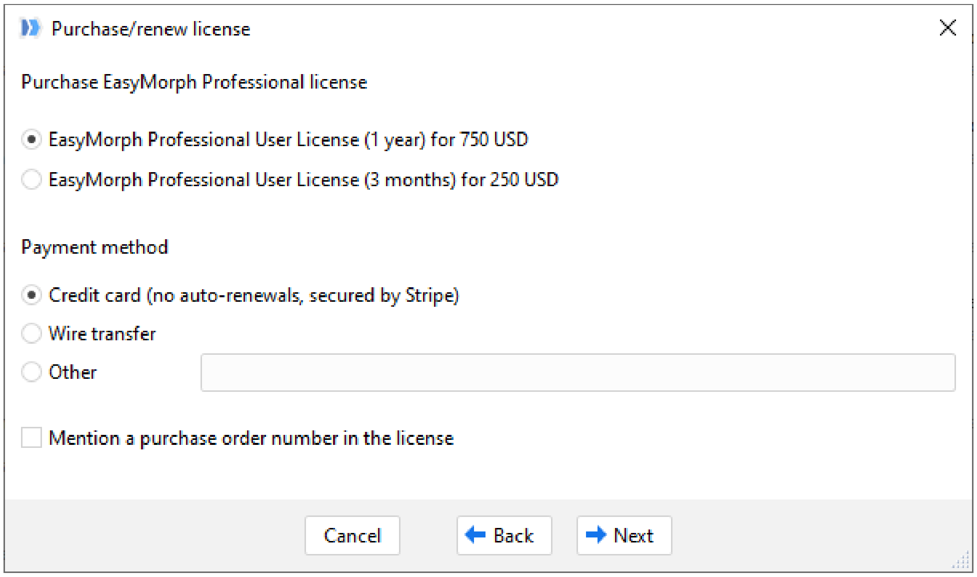
`isinteger(123.45)` returns FALSE.

What's new in Desktop

In-app license purchase wizard

It is now possible to purchase or renew one EasyMorph Professional license automatically right from EasyMorph Desktop. Click the "Buy/Extend license" link on the Start screen of the application, provide order and billing information, and submit order. In most cases your order will be processed automatically: you will be forwarded to a payment web page, and immediately after payment a license key will be sent to you by email, or even installed, automatically.

In some cases your order may require manual processing by our sales team. In this case you will receive a notification with further instructions.



Purchase/renew license

Purchase EasyMorph Professional license

● EasyMorph Professional User License (1 year) for 750 USD
○ EasyMorph Professional User License (3 months) for 250 USD

Payment method

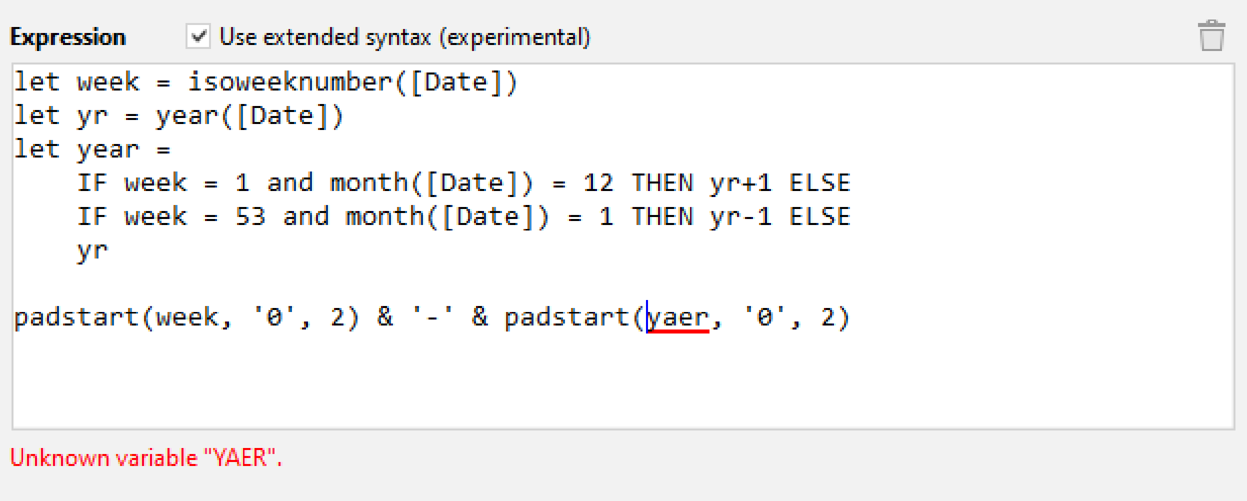
● Credit card (no auto-renewals, secured by Stripe)
○ Wire transfer
○ Other

☐ Mention a purchase order number in the license

Screenshot 7: The license purchase wizard.

Syntax error highlighting

When the extended expression syntax (described above) is enabled, the expression editor highlights the error location. Also, the error message below the text box becomes clickable and brings the cursor to the error location.



Expression ☑ Use extended syntax (experimental)

```
let week = isoweeknumber([Date])
let yr = year([Date])
let year =
    IF week = 1 and month([Date]) = 12 THEN yr+1 ELSE
    IF week = 53 and month([Date]) = 1 THEN yr-1 ELSE
    yr

padstart(week, '0', 2) & '-' & padstart(yaer, '0', 2)
```

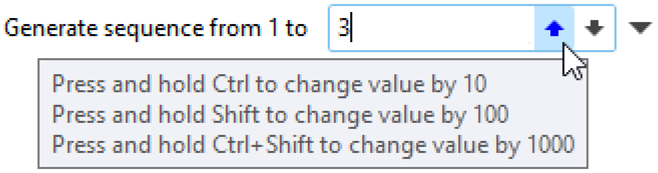
Unknown variable "YAER".

Screenshot 8: Syntax error highlighting.

Usability improvements for numeric input box

Now, it’s possible to increase/decrease numeric settings in steps of 10, 100, or 1000 by depressing Ctrl, Shift, or Ctrl+Shift respectively.

The numeric input box also supports the mouse wheel. The step modifiers also work with the mouse wheel.



Press and hold Ctrl to change value by 10
Press and hold Shift to change value by 100
Press and hold Ctrl+Shift to change value by 1000

Hint: hover the up or down button to see a tooltip with increments.

Miscellaneous

- Derived tables/charts can now be created immediately in a new tab.
- A derived table is now by default named after the table it was derived from.
- Much faster “Aggregate” and “Pivot” actions (typical speedup is 2-3x).
- Faster iterations.
- The “Export to delimited text” action is now faster.
- The “Reload and run” button has moved from the “Project” menu to the “Main” menu.
- A license key can now be installed by simply dragging it into EasyMorph Desktop.
- The “Move to tab” dialog has a new look and the ability to move to a new tab.
- The “Create repository” dialog has a new look and better usability.
- In the Query Editor, aggregations can now be added in the list of fields on the left (right-click a field name).
- In the Query Editor, an aggregation can be applied to multiple selected columns at once.
- In the Query Editor, it is now possible to sort by SQL expressions.
- The Expression editor now shows lists of column names AND parameters. The radio-switch Columns/Parameters has been removed.
- Super-compression is now ON by default in Desktop, Launcher, and Server.

Previous release notes

Link: Release notes for v4.3.1.