Release notes v4.0

Important! Create a backup copy of your projects and connector repositories before updating to the new version. Projects saved in version 4.0 can't be opened in earlier versions.

A few words from the creators of EasyMorph

The 4th version of EasyMorph is the most significant modification of EasyMorph over the last 3 years. It implements a lot of user experience, many feature requests, and effectively brings the application to the next level. Roughly twice better performance, modules, web requests, column lineage, automated data profiling and suggestions – all this makes version 4 a new generation of EasyMorph which is now more powerful, convenient, and secure. Web requests open for EasyMorph a new era of interoperability and data exchange with external services and systems, while preserving the same easy, visual, and interactive authoring that is EasyMorph known for. The new version sets a solid foundation for future development of both EasyMorph Desktop and EasyMorph Server for many years ahead. Many new features of the upcoming releases will be based on the novelties introduced in this version or will extend them in some way.

Another big change is the new approach to the free edition (more on that below). From now on, we pledge that actions created using SDKs and libraries provided by various vendors for free will be available in the free edition as long as implementing them in EasyMorph didn't require significant development effort on our end. For instance, actions "Amazon command" and "Google Drive command" are now available in the free edition. The upcoming integration with Power BI will also be included in the free edition. At the same time, we are reducing the number of actions/iterations available in the free edition to 20/20, which we believe is still not that bad especially taking into account that the free edition offers more than 100 actions, no restrictions on data volume, and the same level of integration with EasyMorph Server as the Professional edition, including the ability to publish and execute projects on Server. Note that previous versions of EasyMorph with higher limits for actions/iterations remain on our website so you can keep using their free editions as long as you want.

Finally, we stepped up the degree of integration between Desktop and Server, tightened connector security, and addressed a few important usability issues with Server voiced earlier by our users. EasyMorph Server has got a new multi-session runtime. While the task management UI looks very much unchanged version in 4.0, the upcoming versions will bring a deep makeover to Server as well as many new features that are now enabled by the new runtime.

We highly recommend reading through all the Release Notes as the changes are numerous.

A special "thank you" goes to our paying customers – without you the new release won't be possible.

Team EasyMorph

Breaking changes

What changed	What can it break
The free edition now allows only 20 actions and 20 iterations. See chapter “What’s new” below for more details.	Projects with more than 20 actions/iterations executed in the free edition will stop working.
Connector names are now case-sensitive. When EasyMorph looks up a connector by name in a connector repository it doesn’t ignore case anymore. For instance, connector called “MyDatabase” now can’t be specified using a parameter which value is “Mydatabase”.	Actions where connector names are specified using parameters may stop working if parameter values ignored case.
Values in column "Error source" generated in Call/Iterate in the Capture errors mode now include the module name.	If you project’s logic depends on contents of error messages generated by Call or Iterate actions, they project may stop working properly.
The “Import from Excel” action now requires explicit indication that data should be loaded from an Excel range.	Projects which have the "Import from Excel" action with source specified as an Excel range will try to load a sheet instead of a range. Such projects should be edited manually to specify a range.
The underlying mechanism for column formatting has changed. Column formats are no longer linked to tables, but instead to column lineage metadata.	Table column width and formatting can be different for renamed columns. This can affect or break formatting in PDF reports.
Import actions (e.g. “Import from delimited text”) now generate an error value instead of a text value for data parsing errors. The message text in the error value is the same as it was in the text values previously.	If your project’s logic depends on contents of error messages, it may stop working properly because cells with error messages now have type Error instead of Text.

Known issues

- Launcher doesn’t write log files. It will be fixed in v4.1.
- Help-articles not available for some new features on the day of release. It will become available later.

What’s new

Re-balanced free edition

The free edition has been re-balanced in version 4.0. It now includes many import/external actions that previously were only available in the Professional edition. All the following actions are now available in the free edition:

- Call
- File transfer (over SFTP/SCP)
- Amazon Command
- Google Drive Command
- Import from Google Sheets
- Export to Google Sheets
- Tableau Server command
- Qlik Sense command
- Split delimited file

Besides that a few restrictions have been removed:

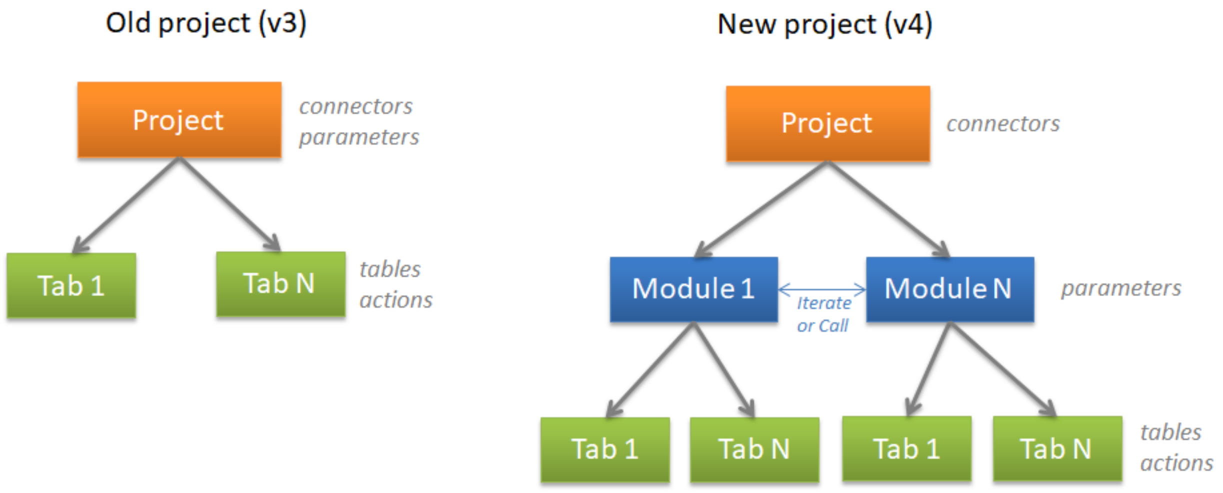
- The “Export to database” action now has no restrictions in the free edition and allows explicit column-to-field mapping.
- The “Database command” action also has its restriction removed and allows using advanced data types in the “Create table” command.

The number of actions/iterations available in the free edition has been reduced to 20/20.

Modules

Projects in EasyMorph now consist of one or more modules. Modules eliminate the need to have a separate project for iterations (loops) or sub-routines. The ability to iterate another project will remain, but it won’t be strictly necessary anymore. In the vast majority of cases iterations are now arranged using modules.

A module is very much what you probably used to think of a project in the previous versions – it consists of tabs (sheets) with tables with actions and can have parameters. The main difference between a module and the old understanding of a project is that modules don’t have embedded connectors. See the illustration below to better understand the difference between version 4 and previous versions:

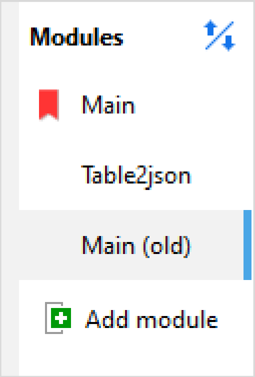


Modules can be used in calls/iterations instead of external projects. By default, the Call/Iterate actions now offer to run a module, not a project.

Parameters are defined for each module individually, not for the entire project. Repository connectors are the same for all modules. Embedded connectors are still specified in project, as it was previously.

Modules are isolated from each other. It's not possible to reference a table in a module from another module. Any module can call/iterate any module in the same project as long as it doesn't create a cyclical dependency. A module can have a default result table, as projects could previously.

In a project there always is a default module. It's marked with a red flag (similarly to default result tables). When a project is executed by another project, or Server/Launcher, the default module is the starting point. The result table of a project is the result table of its default module.



You can switch between modules using the new sidebar on the right (see the screenshot to the right), or by pressing Ctrl+PgUp / Ctrl+PgDn.

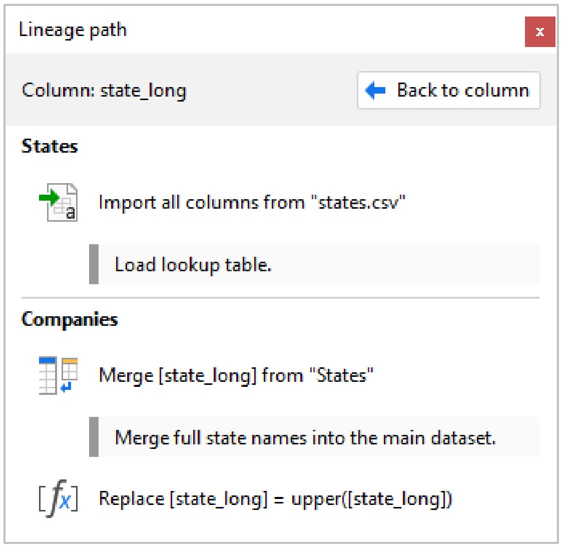
Modules can be cloned, copied/pasted into other projects, or imported from other projects (see menu Design, button "Import module").

A project can have inactive (passive) modules – i.e. modules that are not a starting point and are not called by other modules. It allows for instance keeping older versions of modules around, and reverting back to them in a few clicks, if it becomes necessary.

Column lineage

Column lineage metadata is a new internal mechanism that is used for column formatting, lineage tracking, and storing column metadata (such as annotations to database fields or header cells in Excel).

The lineage path for a column starts at the action where the column was created, and then it tracks the column through all the actions that pass the column further unchanged, or modified in some way. It stops when the column is deleted, or simply ignored (discarded) by the next action. The lineage can track columns from table to table, through derived tables (but not through modules).



Column: state_long

States

Import all columns from "states.csv"

Load lookup table.

Companies

Merge [state_long] from "States"

Merge full state names into the main dataset.

Replace [state_long] = upper([state_long])

Screenshot 1: Column lineage path.

In the screenshot above, column [state_long] was created by an “Import from delimited text” action, then it was merged by a “Merge” action (in another table), and finally it was modified using a “Modify column” action. While the column’s content changed along the path, it’s basically the same column from a semantical perspective.

When user changes column format or width, the new formatting is now applied automatically for all columns along the column lineage path, even if it goes into other tables or the column is renamed on the path.

To see a column’s lineage path, select the column (click its header) and press the “Lineage” button the context bar.

Performance improvements

Due to improvements in algorithms and internal data structures the new version has much better performance in a number of ways:

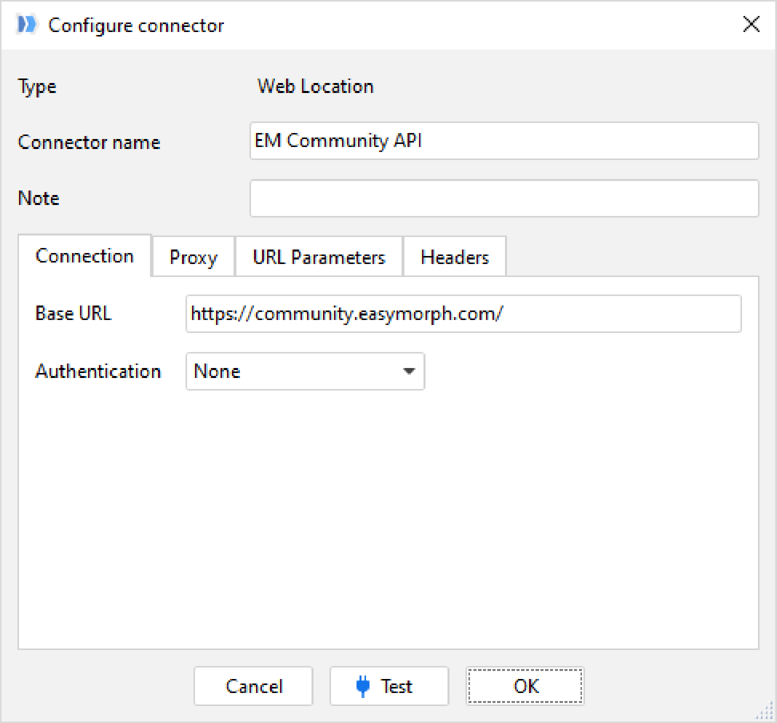
- Faster, more memory-efficient data compression
- Improved algorithms for actions such as Aggregate, Merge, Sort, Deduplicate, Match, Running total, Pivot, Keep/remove matching and some others.
- Up to 4 times faster import from text files
- Up to 3 times faster import from Excel spreadsheets
- Faster import from databases

On average, projects in the new version run 30-50% faster and consume 10-15% less memory than in older versions of EasyMorph.

The "Web location" connector

The "Web location" connector is intended for accessing web APIs using the "Web request" and "Iterate web request" actions (described further). The connector can be configured to automatically insert specified URL parameters or/and HTTP headers into every web request thus simplifying working with APIs that require authorization tokens or/and secret keys to be provided in every request.

Under the hood, for each project run the connector creates and holds a temporary cookie container. Session cookies that come in HTTP responses are automatically placed in the cookie container that is shared with all following web requests during project execution. If a module/subproject is called, the cookie container is silently passed to it, so that the session cookies are available there too. When project execution is finished the cookie container is discarded. This mechanism makes it possible to use the "Web location" connector and the web request actions for cookie-based authorization through emulating web-forms on websites.



Type: Web Location
Connector name: EM Community API
Note:
Base URL: https://community.easymorph.com/
Authentication: None

Screenshot 2: The "Web location" connector.

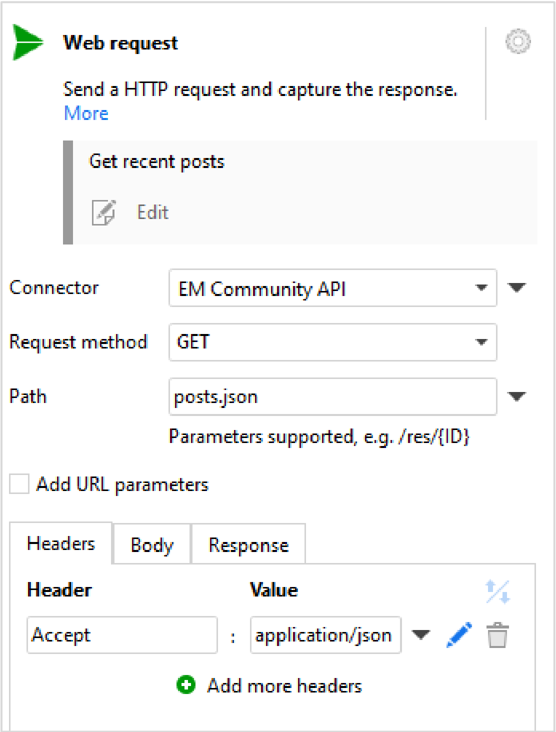
Actions for interaction with web APIs

Two new actions, "Web request" and "Iterate web request" allow sending and receiving HTTP requests from EasyMorph projects without scripting/coding. The "Web request" action sends a single HTTP request. "Iterate web request" sends multiple uniform requests and appends responses into one dataset, if necessary. These are very powerful, albeit somewhat low-level tools for interaction with web

APIs. To use the actions at least a basic understanding of the HTTP protocol is required. The actions provide the following capabilities:

- Constructing a request URL by combining the base URL specified in the “Web location” connector, and an endpoint path specified in the action. Parameter insertion is supported.
- Adding URL parameters specified explicitly, by module parameters, or by column values (Iterate Web Request only).
- All types of HTTP request methods – GET, HEAD, PUT, POST, DELETE, CONNECT, etc.
- Specifying HTTP request headers and their values explicitly, using module parameters, or column values (Iterate Web Request only).
- Construct the request body as a flat JSON, name-value collection of a web-form, contents of a file, free-form text (Web request only), or column value (Iterate Web Request only).
- Obtain response status, headers, and body as column values in the action’s result
- Save the response body into a file

Both actions have the Preview pane that displays the constructed HTTP request, sends it, and displays the response without executing the action.



Web request

Send a HTTP request and capture the response.
More

Get recent posts

Edit

Connector: EM Community API

Request method: GET

Path: posts.json

Parameters supported, e.g. /res/{ID}

☐ Add URL parameters

Headers | Body | Response

Header		Value
Accept	:	application/json

Add more headers

Screenshot 3: The "Web request" action.

To parse JSON/XML use regular actions (see example) and/or the new web functions (described further). Next version (v4.1) will introduce purpose-built actions for constructing and parsing JSON/XML.

Other new actions

The **Repeat rows** action creates duplicates of rows as many times as specified in a given column for each row. A typical use case would be generating a table with a sequence of dates defined by ranges of dates (see example).

The **Enumerate groups** action assigns numbers from 1 to N to each unique combination of values (group) in specified columns. While it was possible to do using a combination of actions previously, the new action makes it simpler.

The **Distribute total** action is somewhat reverse to aggregation. It distributes (breaks down) a total/subtotal using another column's values as weights. A typical use case would be a calculation of projected sales revenue for every product given a total full year projection, and up to date revenue for each product. In this case the projected total is broken down by product. While it was possible to do using a combination of actions previously, the new action makes it simpler.

Changes in existing actions

The **Import from Excel** action now allows importing and automatically appending multiple sheets at once without iterations. New option for adding a column with sheet name.

The **Enumerate rows** action now can enumerate rows inside groups.

The **Keep matching** and **Keep mismatching** actions have been merged into one new action called **Keep/remove matching** that supports multiple key field matching.

The start/end dates in the **Calendar** now can be specified using a parameter.

Queries in the **Import from database** now can use a parameter to specify top-N rows.

Web functions

In addition to the new connector and the new actions for web requests, a number of new functions have been introduced to facilitate web operations:

jsonvalue(JSON_text, JSONPath) – extracts a single value or a JSON object from a node specified by JSONPath. Example:

`jsonvalue("{customer:{ID:345}}", "customer.ID")` returns 345.

xmlvalue(XML_text, XPath) – extracts a single value or an XML node from a node specified by XPath. Namespaces are not supported. Example:

`xmlvalue("<CUSTOMER><ID>345</ID></CUSTOMER>", "CUSTOMER/ID")` returns 345.

isjson(text) – returns true if the given text is a JSON object or JSON array. Example:

`isjson("{customer:{ID:345}}")` returns TRUE.

isxml(text) – returns true if the given text is an XML object. Example:

`isxml("<XML>")` returns FALSE.

combineurl (url, path) – Combines an URL and a URL path into one URL. Example:

`combineurl("https://mysite.com/path1", "path2")` returns https://mysite.com/path1/path2.

hashhex (method, text) – Returns a hexadecimal hash of a text string calculated using one of the available methods (md5, sha1, sha256, sha384, sha512). Example:

`hashhex("md5", "EasyMorph")` returns 530ea21e53be78b34a678373bbda8306.

hmachex (method, text, key) – Returns a hexadecimal HMAC of a text string calculated using one of available methods (md5, sha1, sha256, sha384, sha512). Example:

`hmachex("md5", "Easy", "Morph")` returns a4f47c0d5a1080056a2a560f6832389a.

encode(method, text) – Encodes a text string using one of the available methods ("uriComponentEscape", "Base64", "Base64url"). Example:

`encode("uriComponentEscape", "one+one")` returns one%2Bone.

decode(method, text) – Decodes a text string using one of the available methods ("uriComponentEscape", "Base64", "Base64url"). Example:

`decode("uriComponentEscape", "one%2Bone")` returns one+one.

uriencode(encoding, value) – Encodes a text string using specified charset. Example:

`uriencode("iso-8859-15", "Les Misérables")` returns Les+Mis%e9rables.

uridecode (encoding, value) – Decodes a text string using specified charset. Example:

`uridecode("iso-8859-15", "Les+Mis%e9rables")` returns Les Misérables.

Other new functions

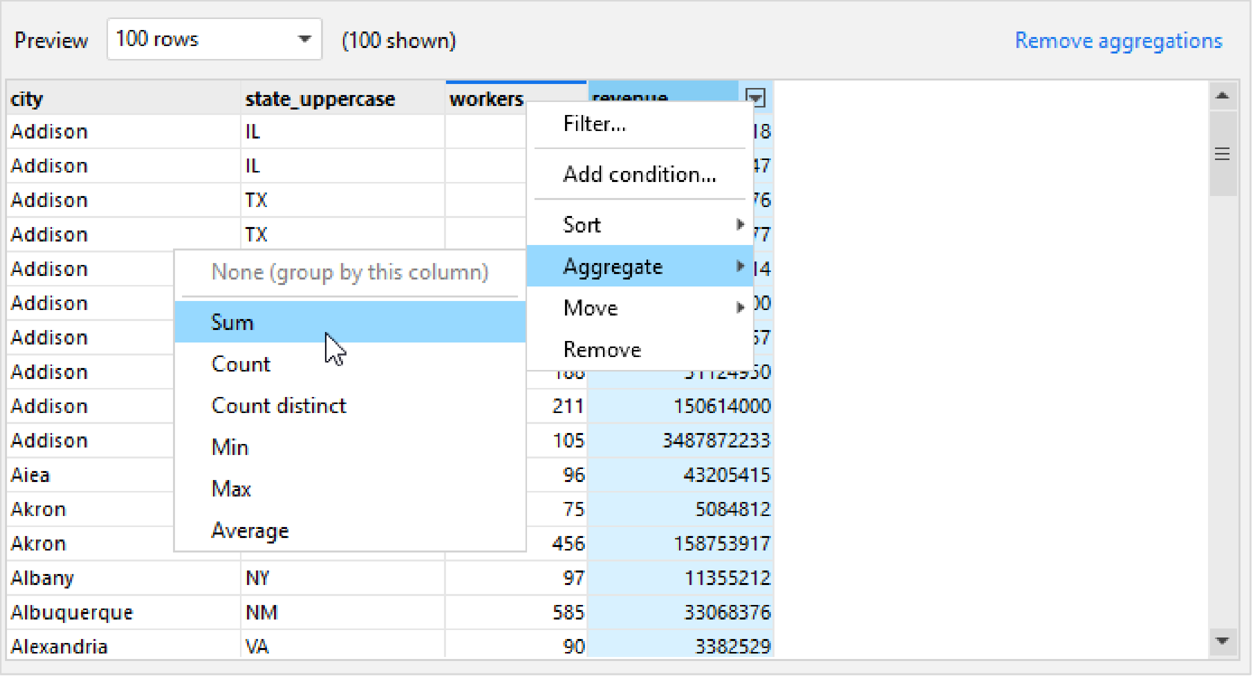
addhours(number_date, number) – Adds the specified number of hours to date.

addyears(number_date, number) – Adds the specified number of years to date.

system('callermodulename') – returns the name of the module that called the current module.

Grouping/aggregation in queries

Queries can now group/aggregate data. The necessary GROUP BY clause will be generated in underlying SQL automatically. To create a grouping/aggregation use the column context menu in the datagrid.



city	state_uppercase	workers	revenue
Albany	NY	97	11355212
Albuquerque	NM	585	33068376
Alexandria	VA	90	3382529

Screenshot 4: Aggregations in queries.

Aggregated fields are marked with a blue line in the field header. All aggregations and groupings can be removed by clicking the "Remove aggregations" link.

Miscellaneous

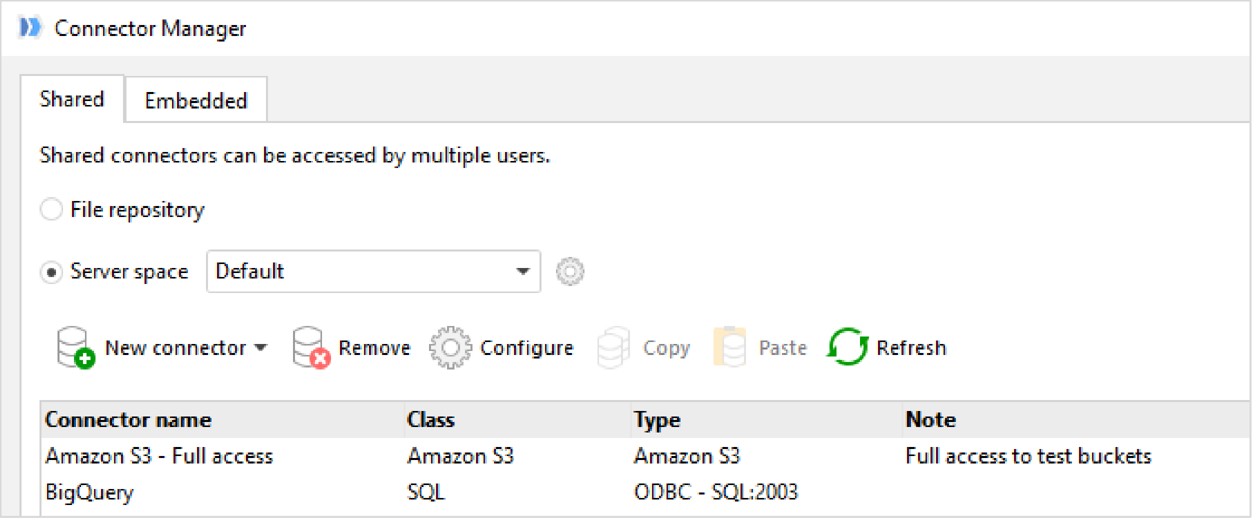
- The "1010data" connector has been deprecated.

What's new in Desktop

Server-hosted connector repositories

With the new version it is now possible to use and edit connectors of a Server space right from Desktop. To use a Server-hosted repository, switch to it in the Connector Manager (see screenshot below). Note that using connectors on Desktops is disabled by default and should be enabled in the space security settings.

Working with Server-hosted repositories is no different from working with file repositories. You can create new connectors, edit existing connectors, copy/paste connectors, and set/remove a write password for the repository.



Shared connectors can be accessed by multiple users.

○ File repository

● Server space Default

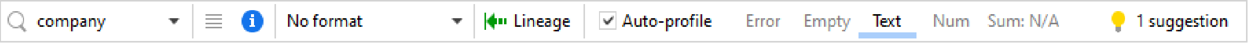
Connector name	Class	Type	Note
Amazon S3 - Full access	Amazon S3	Amazon S3	Full access to test buckets
BigQuery	SQL	ODBC - SQL:2003	

Screenshot 5: Server-hosted repository in Desktop.

As with file-based repositories, Launcher always uses the same repository as Desktop.

Context bar

The new context bar is intended for column search, formatting and instant profiling.



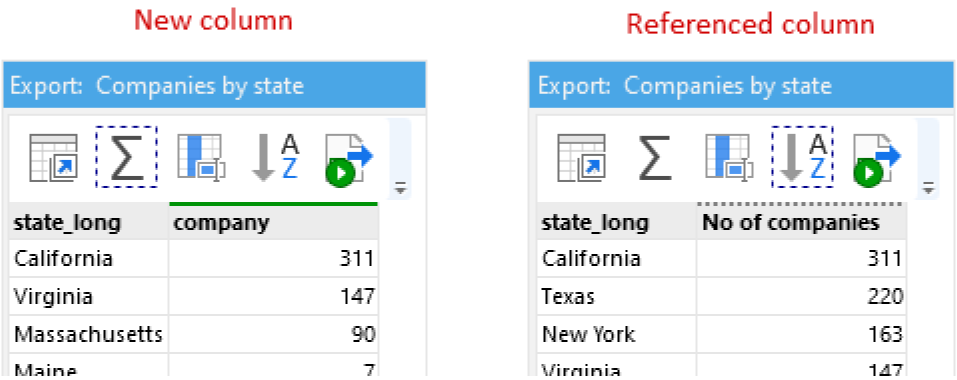
Screenshot 6: Context bar.

When a column is selected, the context bar displays:

- Column search box
- Button that opens the “Column profiler” dialog with the list of unique values and other metadata
- Indicator of column annotations
- Column format selector (works as well with multiple columns selected)
- Button “Lineage” to open the “Lineage path” dialog
- Data types of column values with counts for each type (hover to see)
- Quick aggregation such as sum, min, max, or average
- Data quality suggestions (if any)

Indication of new and referenced columns

The datagrid in tables now indicates columns that have been created by the action as well as not new (i.e. created in a previous action) columns that are used (referenced) in column properties. New columns are displayed with a solid green line in the column header. Referenced columns are indicated with a dark grey dotted line.



state_long	company
California	311
Virginia	147
Massachusetts	90
Maine	7

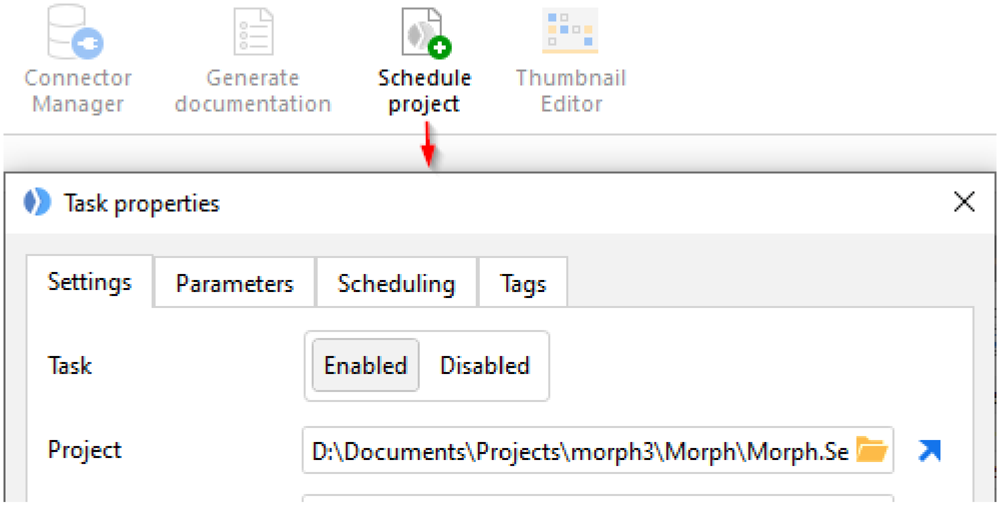
state_long	No of companies
California	311
Texas	220
New York	163
Virginia	147

Screenshot 7: Indication of new and referenced columns.

Note that actions that support grouping by one or more columns do not mark such columns as referenced.

Scheduling projects in Launcher right from Desktop

It is now possible to schedule a project in Launcher right from Desktop. Go to menu “About” and press the “Schedule project” button. A new Launcher task will be created and scheduled for execution. Note that Launcher should be running in order for this feature to work.



Task: Enabled Disabled

Project: D:\Documents\Projects\morph3\Morph\Morph.Se

Screenshot 8: Scheduling a project from Desktop.

Task tags in Launcher

Launcher tasks now can be tagged with one or many tags. The list of tags can be used to quickly filter tasks. It is possible to filter by multiple tags at once – use *Ctrl+click* to select multiple tags for filtering.

Manage tags

Tags are created and edited in the “Tag manager” dialog.

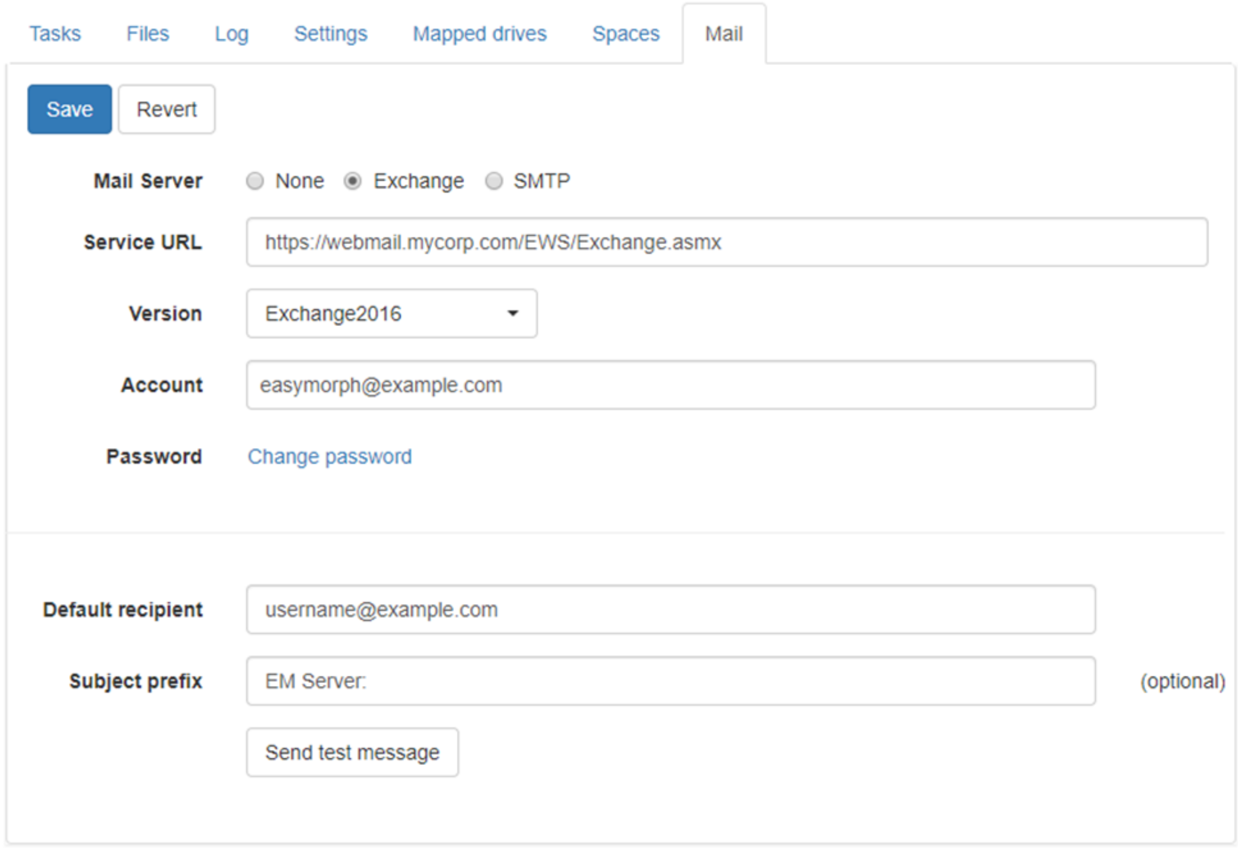
Miscellaneous in Desktop/Launcher

- New datagrid cell/column selection style
- New tab “Metadata” in the Column Profiler
- New data quality suggestions
- With an invalid/expired license Desktop works in the free edition mode without resetting key
- New shortcut: *Ctrl+double click* table title bar to minimize the table
- Collapsible left/right sidebars
- Launcher now automatically reloads connectors when they are changed in Desktop
- Launcher now has the "Reload connectors" button for manual repository reloads
- Watermark helpers in connectors

What’s new in Server

Email notifications about failed tasks

EasyMorph now can send email notifications about scheduled tasks that failed. Email settings are configured in the new Mail tab available in the Server settings. Email notifications can be sent via Microsoft Exchange or any IMAP email server.



Mail Server ○ None ● Exchange ○ SMTP

Service URL https://webmail.mycorp.com/EWS/Exchange.asmx

Version Exchange2016

Account easymorph@example.com

Password Change password

Default recipient username@example.com

Subject prefix EM Server: (optional)

Screenshot 9: Email notification settings in EasyMorph Server

Additional space security settings

The space security page now shows a few new options:

- "Repository access for Desktop users" – disable, read-only, or full access to the connector repository from Desktops users.
- Allow/forbid copying connectors to clipboard by Desktop users.
- Allow/forbid arbitrary code execution on Server. Disables/enables actions "Run program", "Iterate program", "PowerShell", and "SSH command".

Miscellaneous

- New "Upload" button in Sever Settings for uploading a license key
- New "Upload" button in Space Settings for uploading a repository file
- Option for creating a new repository when a space is created

Previous release notes

Link: Release notes for v3.9.x.