EasyMorph Hub Administrator Guide

Version 6.1.0 September 4th, 2026

Table of Contents

About EasyMorph Hub

EasyMorph Hub is a locally installed (i.e., on premises or virtual machine) Windows application. It is installed and operates as a Windows service with a web UI and is accessible via a web browser.

EasyMorph Hub is used for the following:

- Running EasyMorph workflows designed in EasyMorph Desktop
- Browsing local and remote folders, uploading and downloading files
- User collaboration on workflows, files, metrics, datasets, and various data assets
- Storing connection settings for workflows
- Managing licenses of Desktop users
- User activity logging

Prerequisites

- 64-bit version of Windows 10 (or above) or Windows Server 2019 (or above)
- .NET 8 (if missing, it will be installed automatically by the Hub installer)
- MSVC++ 2015 Redistributable Packages (usually already installed, if not then download here)
- Tableau drivers require MSVC++ 2013 Redistributable Packages (usually already installed, if not then download here)
- 200 MB free disk space

Memory requirements

EasyMorph Hub is a memory-intense application because it processes all data in memory. The required amount of RAM greatly depends on volumes and characteristics of data processed by Hub workflows. Therefore, estimating exact memory requirements is not straight forward. Due to aggressive on-the-fly data compression in EasyMorph, the amount of consumed RAM highly depends on data types and cardinality. A very (very!) rough rule of thumb is 4GB plus 1GB per each 1 million rows in the largest dataset to be loaded entirely in a workflow. Add up dataset sizes for simultaneously executed workflows.

It's possible to use EasyMorph Desktop to obtain memory consumption estimates because both EasyMorph Hub and EasyMorph Desktop employ exactly the same in-memory engine under the hood. Run actual workflows with real data in EasyMorph Desktop first and track its memory consumption in Windows Task Manager. Note that EasyMorph Hub typically requires up to 10-20% less memory than Desktop due to server-specific optimizations.

What would happen if the Hub runs out of memory?

Briefly speaking, it's highly undesirable. The system would become unstable and would behave unpredictably. In the best case, depending on system settings Windows will try to allocate more memory by swapping parts of RAM to disk. This will drastically slow down workflow execution, delay scheduled

tasks and possibly break schedule frequency, and can even make the Hub unresponsive for a period of time. In the worst case, workflow execution will fail producing the “Out of memory” error.

When processing large amounts of data that don't fit into memory, design your workflows so that they partition and process your data chunk by chunk (e.g. day by day) using iterations (loops). EasyMorph has all the necessary means for partitioning data in large database tables and text files, so you can process terabytes of data and billions of rows without loading everything at once.

Note that memory is only consumed when a task is running. As soon as the task is finished, all the memory used by the task is freed up. When idle, the memory footprint of the Hub is insignificant (a few tens MBs) and constant.

It’s recommended to have a reserve of memory to avoid running out of it. To help detect possible memory deficits, EasyMorph Hub logs warnings into the Hub log when available RAM falls below 20% and sends email notifications. Also, the tab "Task" of the Hub's web-console and the administrative tabs display a live indicator of available memory.

Besides that, you can configure soft and hard memory limits for Hub worker to control memory consumption (described further in this guide).

Quick start: Installation and initial configuration

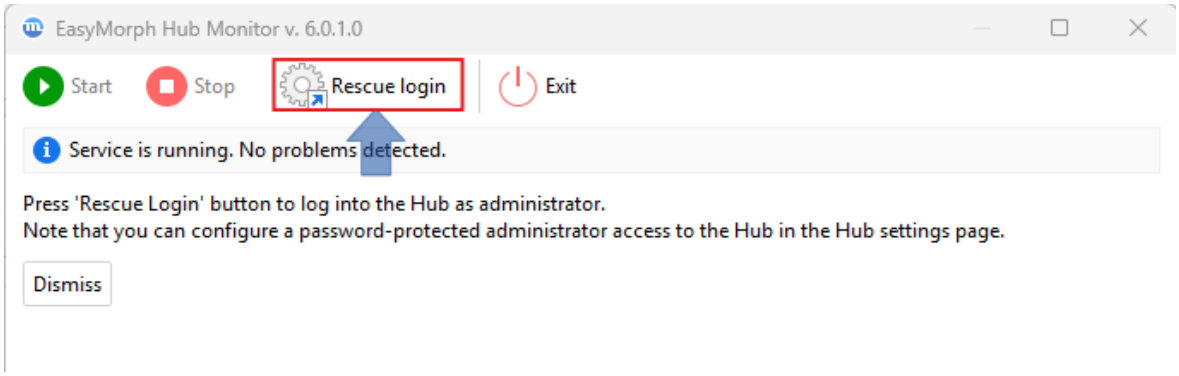
Download EasyMorph Hub installer from the website: https://easymorph.com/download-server.html.

Run the Hub installer under a Windows administrator account and follow instructions. The installer installs the following:

- EasyMorph Hub service (Morph.Server.WebConsole.exe)
- EasyMorph Hub Monitor application (Morph.Server.Monitor.exe)

First login

After successful installation, EasyMorph Hub starts automatically. Also, the Hub Monitor, a Hub configuration utility described in detail further in this Guide (see "Hub Monitor"), is launched simultaneously.



Screenshot 1: The "Rescue login" button in Hub Monitor.

For the initial logging in, press the "Rescue login" button on the toolbar of Hub Monitor (see screenshot above). This will open your web browser with the following URL: http://localhost:6330/settings/edit. The "rescue login" allows accessing Hub as an administrator (from the local Windows group "Administrators") without a password on the computer where EasyMorph Hub is installed.

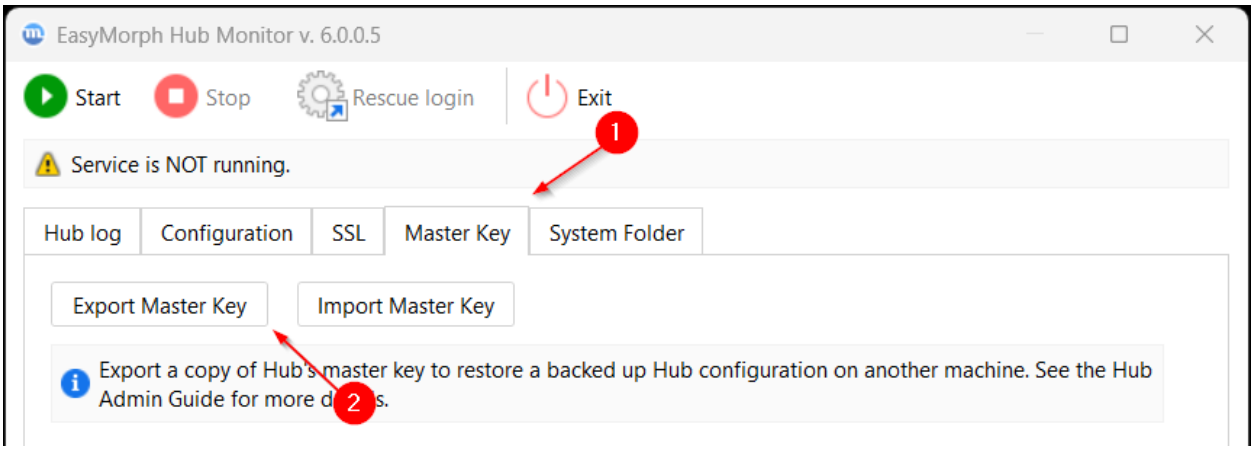
Note that logging in using the "rescue login" automatically creates a Hub user connected to that authentication method (identity provider "Local Windows Administrators"). You can allow Hub administrators to log in via web browsers instead of using the "rescue login" in Hub Monitor as follows:

1. In the user settings, configure a password for that user or link an external identity (such as AD account).
2. Check “The user is a server administrator also when logged in from a web browser” in the user settings, tab “Access”

Master key

It is highly recommended to create a secure copy of the master key of your Hub installation. They master key is required for restoring a Hub installation from a backup on another machine, or for migrating a Hub installation to another machine. Never store the master key on the same machine where EasyMorph Hub is installed.

The master key can be exported in EasyMorph Hub Monitor (see below).

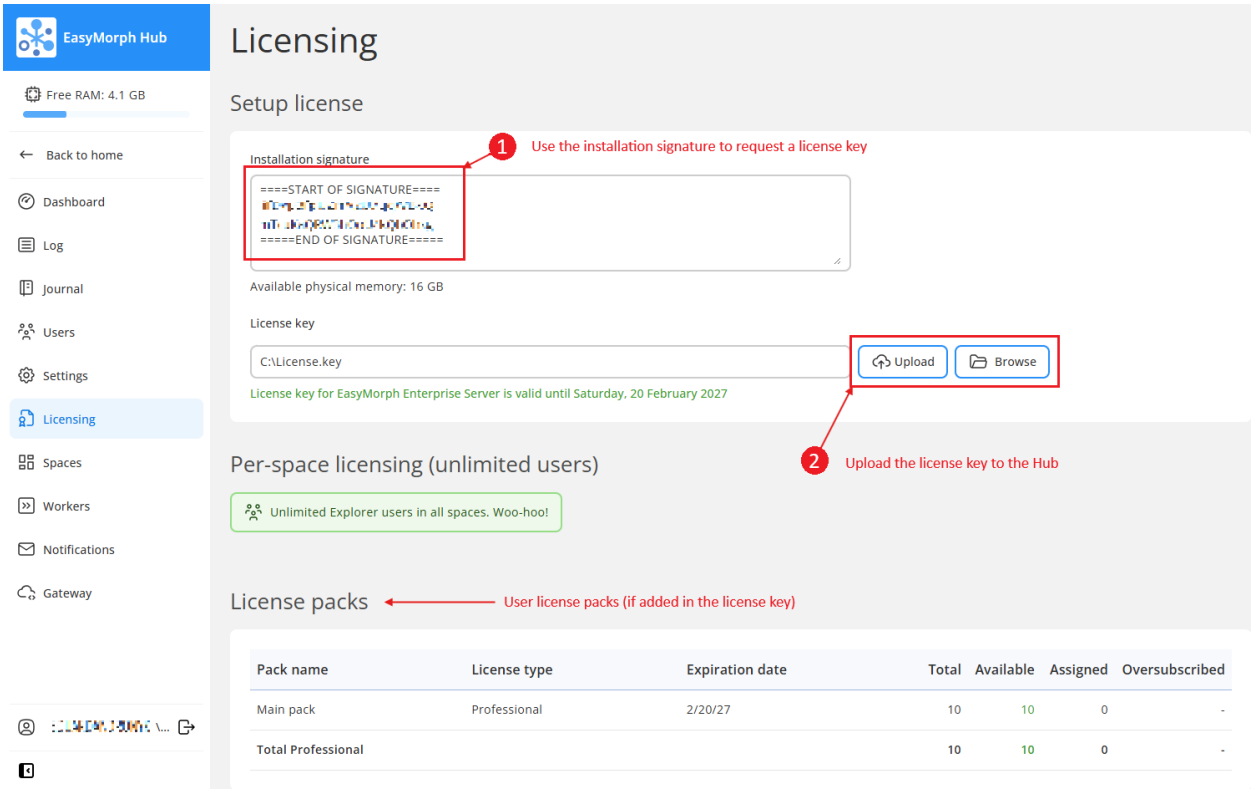


Screenshot 2: Exporting the master key in the Hub Monitor.

License key

Without a valid license key, the Hub doesn't operate normally. To configure the license key:

1. Send the installation signature from the Licensing tab (see the screenshot below) to sales@easymorph.com to request a license key for the Hub, if you don’t have it yet.
2. Upload the license key to the Hub using the "Upload..." button. If the key is valid, it will be applied immediately. No Hub restart is required.



Screenshot 3: License key configuration.

A license key may have a mix of Hub and Desktop (Professional) licenses. The contents of the key can be seen in the license certificate (a PDF file) that comes with the license key. Desktop user licenses can also be assigned right from Hub's web-console. See the chapter "Managing Desktop user licenses" further in this guide for more details.

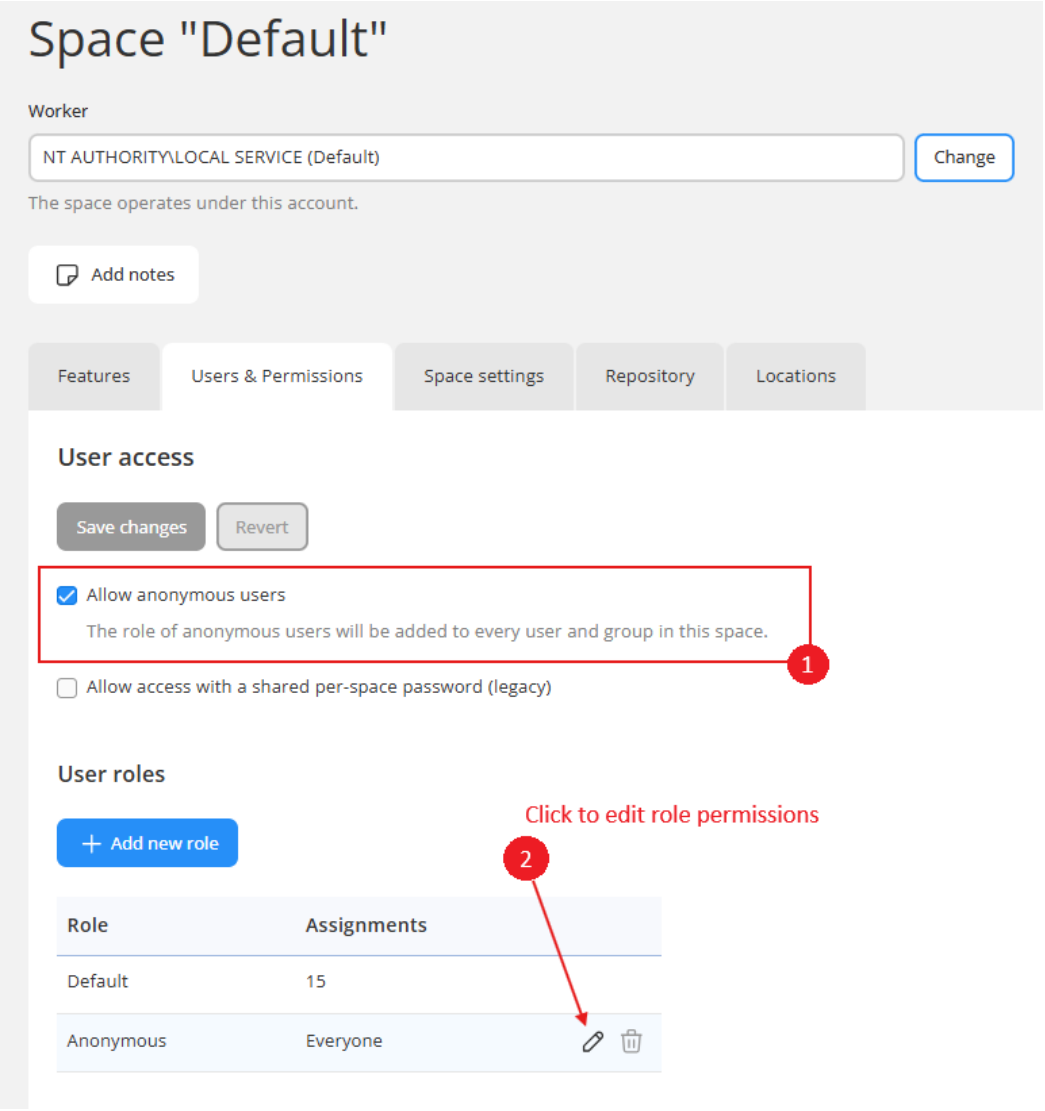
Note that when computer name or domain name changes, the Hub signature also changes. Therefore, if you migrate your Hub to a different machine, contact sales@easymorph.com in order to transfer your license to another computer. You can also request a 30-day temporary (trial) key for testing a new deployment.

Initial user access

By default, no one else can access the Hub except for the Hub administrator who has logged in initially. To enable access for business users, the administrator must enable user access and configure user roles. EasyMorph Hub has a flexible role-based account management system explained later in this Guide (see "Managing user permissions in spaces").

For a quick start, you can use the simplest (but less secure) configuration by allowing anonymous user access to the Default space without authentication:

1. Go to the settings of the Default space and enable Anonymous access.
2. In the list of roles, click the Anonymous role and enable all permissions of the Anonymous role.



Screenshot 4: Configuring Anonymous access to space.

Now any business user can access the Default space without a Hub user account by opening in a web-browser the following link:

http://<your_Hub_host>:6330

In the future, you can configure individual user accounts and groups (such as Active Directory groups) for controlling user access to the Hub.

Creating Hub user accounts and passwords for them is described further in this guide.

It is highly recommended to disable anonymous access to the Hub after you become more familiar with Hub settings. Remember that anonymous access is only enabled by default to simplify the initial Hub configuration.

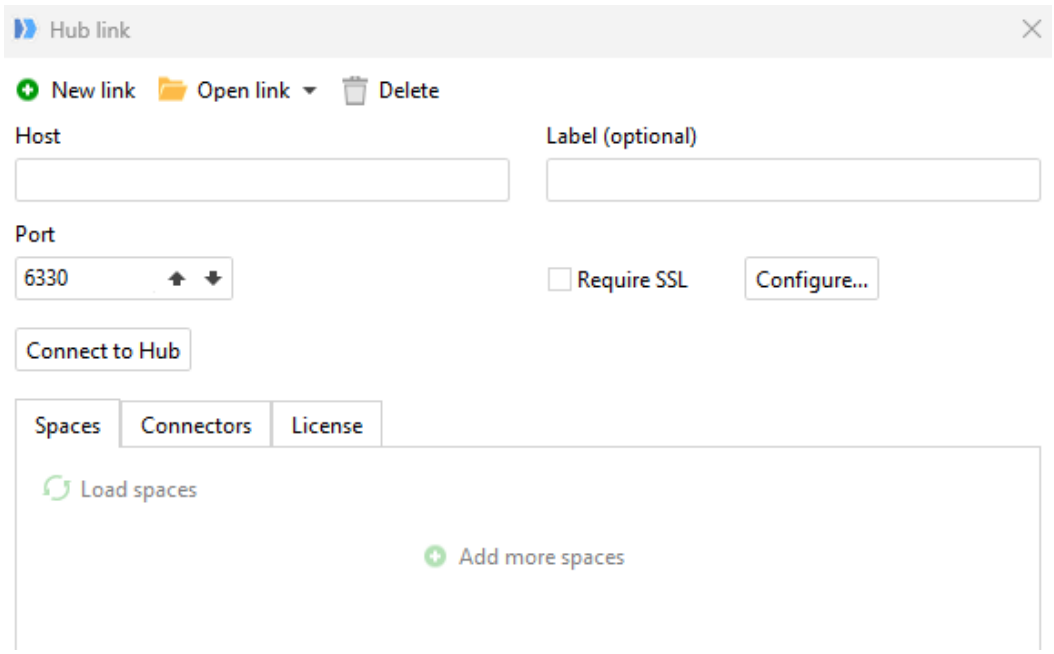
Using Active Directory or Microsoft Entra ID (Enterprise edition only)

If user authentication is expected to be done via Active Directory or Microsoft Entra ID instead of anonymous access, configure the Active Directory connection in the menu "Settings", the tab “Identity providers” and then add the necessary user accounts and groups in the space access control list (ACL) in the tab "Users & Permissions" (can be seen in the screenshot above).

Hub Link in Desktops

For convenience, users that create or edit EasyMorph workflows in EasyMorph Desktop can configure the Hub Link on the Home screen of EasyMorph Desktop. When the Hub Link is configured, users can publish and collaborate on workflows, data connectors, and datasets right from EasyMorph Desktop.

It is highly recommended to have Hub Link configured in Desktops for all users.



Screenshot 5: Hub Link in EasyMorph Desktop.

Updating from previous versions

To update from a previous version:

1) Stop the EasyMorph Server service
2) If the service is installed under a different account than *NT AUTHORITY\LocalService* then write down that account and make sure you know its password.
3) Check the Release Notes if it contains a clear demand for uninstalling the previous version before installing the new one. If it does not (which is typical) then skip to the next step. If it does, uninstall EasyMorph Hub. Do not uninstall the previous version unless it explicitly requested in the Release Notes.
4) Install the new version. In the installer choose “Use existing configuration” (selected by default).

5) The Server service will be installed under the default account (LocalService). If your EasyMorph Hub previously used a non-default account, change the “Log On” account in the EasyMorph Server service properties in Windows Services panel.
6) Start the Server service using either the Hub Monitor (see "Hub Monitor"), or the Windows Services panel.

Hub users

Anyone can access the Hub as a *user,* or anonymously (only when anonymous access is enabled in the space settings). Hub users and their accounts are managed in the tab "Users" available only to Hub administrators.

Hub administrators

For anyone who logs in using the "rescue login" in Hub Monitor, a Hub user is created automatically with the permissions of Hub administrator, including:

- Access to the administrator-only tabs of Hub's UI
- The ability to permit/forbid another user (or himself/herself) to be an administrator when logged in normally, through the Hub's web UI (instead of "rescue login"). This option is configured in the user settings, tab "Access".

Hub administrators can change any Hub or space settings, including user roles and their permissions. Therefore, it is recommended to restrict the number of people who can be Hub administrators.

Note that Hub administrators by default can't access user-created content in spaces (e.g., Hub tasks) unless they are explicitly added to the access control list (further, ACL) of space and are assigned one of the user roles configured in the space.

Anonymous users can't be administrators.

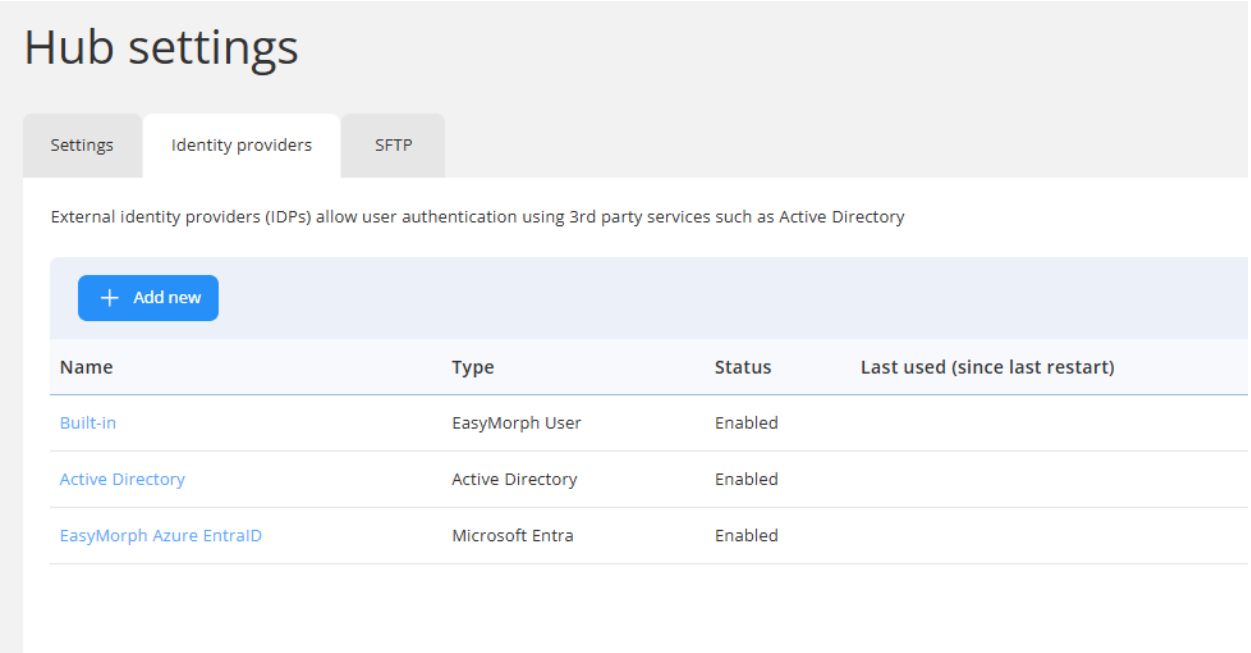
User accounts

For a user to access the Hub, a user account must be created in the Hub and a login method (identity provider) must be configured for that account. Typically, it happens in one of these two ways:

- Automatically, when an Active Directory group is configured in the ACL of a space, and the user is a member of the group
- Explicitly, when a Hub administrator manually creates a user account in the tab "Users" and configures a password for that user. Note that Hub users can't change their passwords themselves, only administrators can set, change, and disable user passwords. Configuring a password is optional. Users can sign into Hub without a password using an external identity (explained in the next paragraph).

Identity providers

EasyMorph Hub has its own simple system of user accounts (the built-in identity provider). Besides that, it supports linking user identities to external identity providers for user authentication. A Hub user can have multiple external identities linked to its Hub account, and thus use multiple methods (providers) for authentication.



Screenshot 6: Identity providers.

Currently, the following identity providers are supported:

- Active Directory
- Microsoft Entra ID
- OpenID Connect (with support for Keycloak roles as groups)

After an external IDP is configured, it can be used for seamless user authentication. To use an external identity, add it in the user settings or in the space ACL. If the IDP supports user groups, they can also be used for user authentication.

Using Active Directory as identity provider

Configure user account and password in the Active Directory settings in the Hub settings.

Important! AD groups must be of the "Security" type (configured in Windows AD group properties) to be authenticated by EasyMorph Hub. If a group's type is set to "Distribution", the group can't be used for user authentication in EasyMorph Hub.

Using Microsoft Entra ID as identity provider

Create an Azure app with the following scopes:

- Type: Application
  - Group.Read.All - Read all groups
  - User.Read.All - Read all users' full profiles
- Type Delegated
  - User.Read - Sign in and read user profile

User groups

External user groups, such as Active Directory or Microsoft Entra groups, can also be configured and used for user authentication in Hub spaces.

When a user is authenticated via an external group, a Hub user account is created automatically and linked to the external identity provider of the group. A password or an external identity can be later added to that account manually by a Hub administrator, but that's optional.

Note that the built-in identity provider doesn't have user groups.

User roles and permissions

User permissions are defined by *user roles* that are configured for each space individually. There are no Hub-wide user roles applicable in all spaces. See the next chapter for more details.

User suspension and deletion

A user account can be *suspended*. In this case, the user account remains but it's no longer active, i.e., the user can't log in or perform any operations with the Hub. A suspended account can be re-activated.

A user account can also be *deleted*. User deletion is permanent and can't be reversed.

Spaces and their configuration

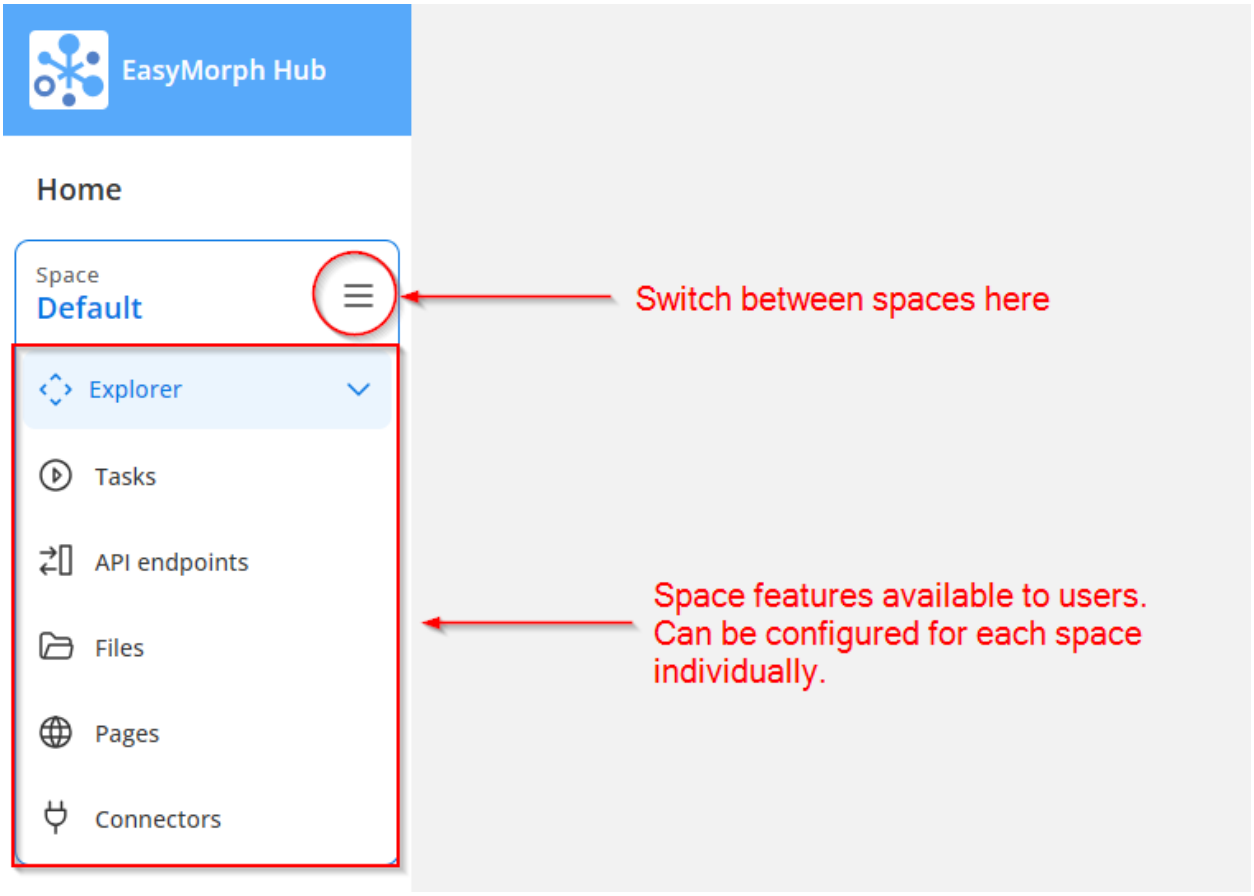
What is a space

A Hub space is a workspace for a team of users. The purpose of spaces is to separate and manage access to tasks, assets, and files for different user audiences. Each space is isolated from other spaces, and has individually configured:

- Space features (e.g. Tasks, Files, or Explorer)
- User access control list (ACL)
- Public folder (accessible through a web-browser or by other means, e.g. via API)
- Hub worker (a child Windows process that is used to run tasks and access files in the space)
- Connector repository and its availability to Desktop users
- Security settings

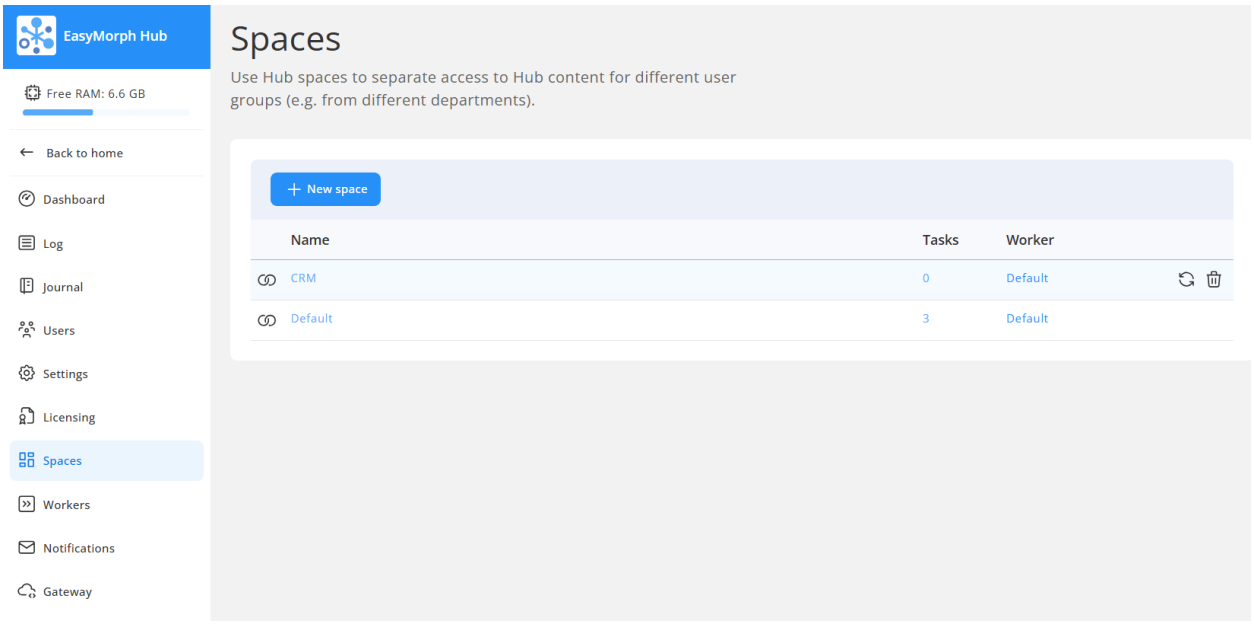
Typically, when an organization uses EasyMorph Hub, spaces are assigned to various departments of that organization. For instance, the finance department can have its own space. The marketing department – its own, and so on.

Space features are available to users can be seen in the left sidebar menu (see below):



Screenshot 7: Space features in the sidebar menu.

Spaces and their features are configured in the tab called “Spaces” visible only to Hub administrators.



Screenshot 8: Hub spaces.

Configuration of user roles and permissions in spaces and space features are explained in detail in respective dedicated chapters further in the guide.

The Default space

The Default space is created automatically during the initial Hub installation. It must not be removed or renamed. If the Default space is missing then it will be created automatically again.

In the Basic and Starter editions of EasyMorph Hub, the Default space is the only available space.

Note that if an external application makes requests to Hub's API and omits space name, it is implicitly assumed that it's related to the Default space.

The Public folder

The public folder of a space is the folder that stores user files such as published workflows, generated spreadsheets, source text files, or datasets. The public folder is meant for:

- Storing EasyMorph projects (.morph files) used by Hub tasks, Catalog assets, and API endpoints
- Storing/sharing EasyMorph datasets (.dset files)
- Collecting data files required for Hub tasks
- Publishing/sharing output files produced by Hub tasks, Catalog or API workflows
- Sharing EasyMorph projects between users

For convenience, it is recommended to store EasyMorph projects in a folder structure with a common root folder, separately from data files. Storing all projects under a common root folder is convenient for performing backup/restore or using a 3rd party version control system (such as git). It also allows using relative paths to refer to input data and helper files, which is convenient for publishing projects from a user's computer to Hub.

If the feature "File Manager" is enabled in the space settings, then it becomes possible to browse, upload, and download files in the tab "Files" using a web-browser. This feature is described in more detail further in this guide (see "File Manager").

Note that two spaces can technically point to the same public folder, or have nested public folders (i.e. the public folder of one space is a subfolder of the public folder of another space).

The public folder must have Windows permissions that permit the Hub service to access the folder. By default, the Hub service runs under the *NT AUTHORITY\Local service* account. Even when a user is logged in using an Active Directory account, accessing the public folder is still done via the Hub's service account.

The Enterprise edition of EasyMorph Hub allows configuring *workers* – additional Windows processes that can run spaces under Windows accounts other than the one used by the Default worker. In this case, the Windows permissions of the public folder of a space must allow the worker process to access the folder. To learn more about workers, read the chapter "Workers".

The “EasyMorph Hub Command” action allows uploading/downloading files to/from a public folder (or its subfolders) from EasyMorph projects executed on desktops or other Hubs. No additional software installation is required.

Also, external applications can upload/download files to public folders using EasyMorph Hub CLI and .NET SDK.

File locations in workflows

To specify paths to data files, it is recommended to use a project/task parameter for the root data folder and calculated parameters for paths to specific files used in the project. In this case, you can publish projects to Hub from your Desktop, and then specify the public folder as the root folder path parameter in the Hub task properties.

For instance, if you design an EasyMorph project on your local computer and it needs to read *C:\My documents\MyData\myfile.csv* you can create two parameters:

1. Parameter {Data root} = C:\My documents\
2. Calculated parameter {CSV file} = combinepath({Data root}, '\myfile.csv')

To specify the file location in an import action, use the parameter {CSV file} instead of a hardcoded path. After the project is published to EasyMorph Hub, create a task and in task properties specify the data root parameter using a path to a Hub folder (e.g. the public folder):

{Data root} = M:\SharedDataRoot

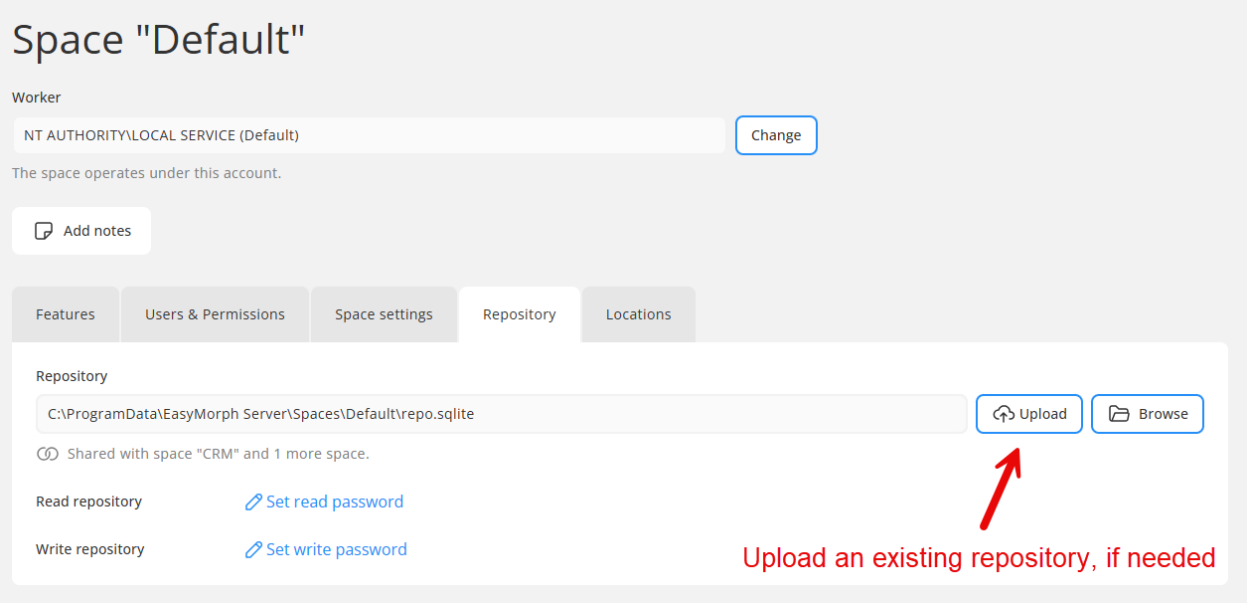
When the Hub runs the task, it will read the CSV file from *M:\SharedDataRoot\MyData\myfile.csv* instead of the location specified in the project (i.e. *C:\My documents\MyData\myfile.csv).*

Connector repository

Each space has a *connector repository* that stores data connectors (e.g., database, email, and API connectors) used in workflows that run in that space.

Technically, the repository is an encrypted SQLite file located in the system folder. Note that the repository doesn't store the actual data from external systems. It only stores the connector settings such as connection strings or OAuth tokens. The repository also stores the contents of Shared Memory, a built-in key-value store in EasyMorph.

The initial EasyMorph Hub installation comes with an empty connector repository in the Default space. You can see the path to the repository in the space settings page. To add, edit, and delete connectors in the repository use the Connector Manager of EasyMorph Desktop (the Hub Link must be configured in the Desktop).



Screenshot 9: Connector repository configuration in space settings.

A new space, by default, points to the repository of the Default space. During space creation, you can choose to create a new repository for the new space or use a repository of another space.

EasyMorph Hub uses the same repository format as EasyMorph Desktop. Therefore, if a repository was created with EasyMorph Desktop it can be uploaded to Hub and used by Hub workflows.

From a security standpoint, it is highly recommended to use Hub-hosted repositories on Desktops as well. In this case, the repository file is not accessible directly from Desktop. Instead, the Hub provides connectors to Desktops on demand.

Important! Do not share the same repository file between Hub and Desktops via a shared network folder. The Hub coordinates access permissions and resolves access conflicts. However, if the repository file is accessed as a file directly by some Desktop, this will interfere with Hub operation and can cause a database file lock.

Repository encryption

Since repositories can contain connectors to sensitive data and the connector settings can include credentials and API keys, all repositories are encrypted using an industry-standard 2048-bit encryption algorithm.

For better protection, it's possible to additionally configure read and write passwords for repositories.

The read password is required for retrieving connectors from a repository and is configured in space settings. Desktop users don't need to provide a read password to access connectors in a Hub-hosted repository.

The write password, when set, must be provided by users when they need to create a new connector, or update/delete an existing connector from EasyMorph Desktop.

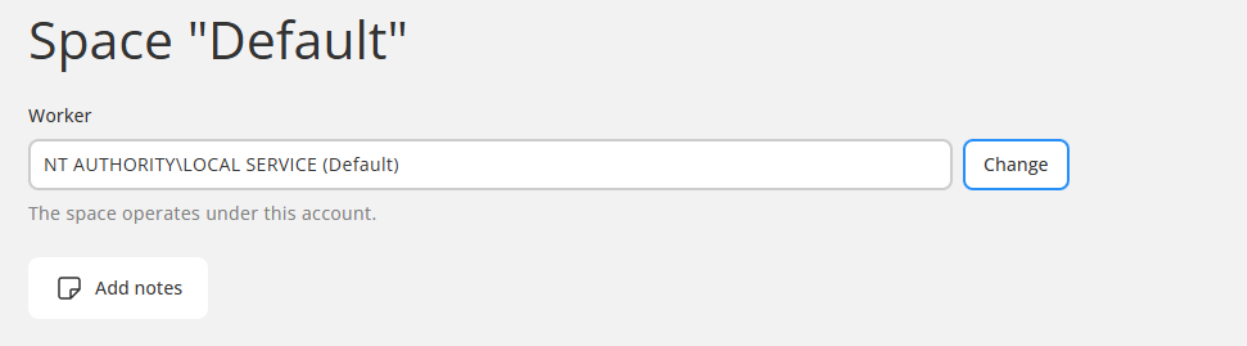
Shared repositories

Two spaces can refer to the same repository file. It is recommended to keep the repository file outside of the public folder in order to restrict user access to it.

Do not share a Hub repository with another active Hub or with Desktops over a network folder. Access to repository is managed by the Hub. If the repository is accessed by another application it may lead to access violation and workflow errors.

Space worker

By default, a space runs tasks and accesses files under the Windows account that is used to run the Server service (see "Server service account" for more details). It's called the Default worker.



Screenshot 10: Worker that performs space operations.

In the Enterprise Edition, it is possible to create workers that use another Windows accounts and assign them to spaces. Such spaces will run tasks and access files using another worker's Windows account instead of the Server service account (the Default worker) which is more preferable from a security endpoint. For more information read the chapter "Workers" further in the guide.

Arbitrary code execution

Workflows in EasyMorph projects can have actions that execute arbitrary code (e.g. a script or an application), specifically, the following actions:

- Run program
- Iterate program
- PowerShell
- SSH Command
- Call Python

Under certain conditions this can be undesirable from a security standpoint, especially when projects are executed under the built-in Default worker (that can access system files of Hub). The "Arbitrary code execution" setting can be used to disable execution of such actions in projects (it's disabled by default).

Execution of unsigned projects

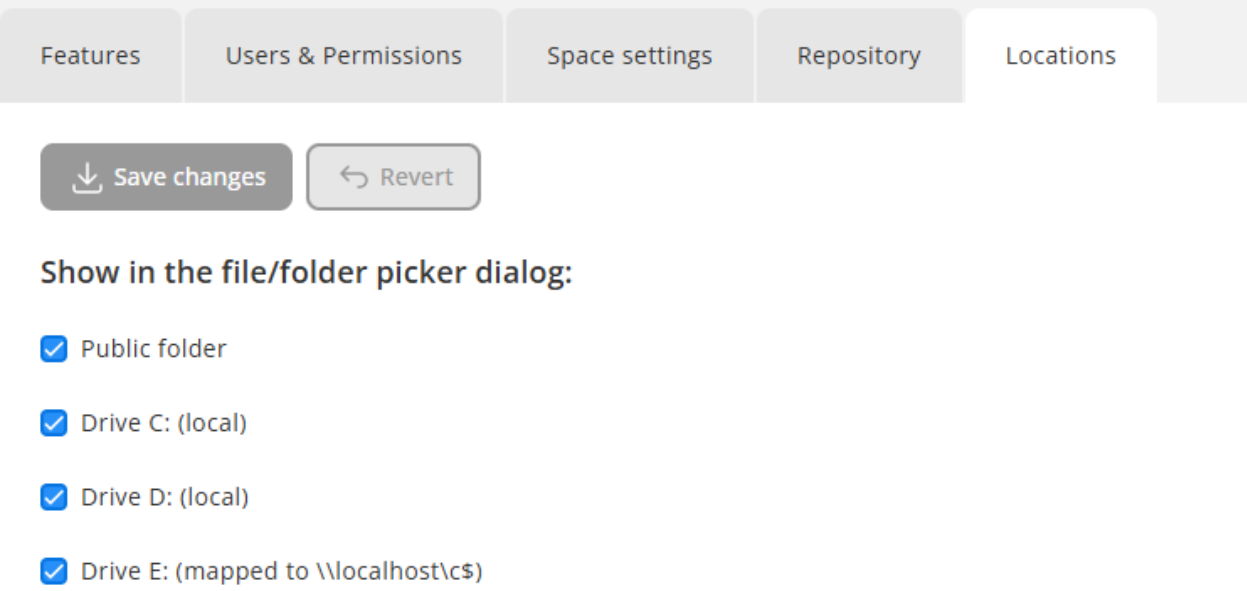
Projects edited in EasyMorph Desktop are digitally signed. If a project is modified outside of EasyMorph (i.e., in a 3rd party application), its digital signature becomes invalid as only EasyMorph Desktop can digitally sign its projects. When EasyMorph Hub executes a project, it verifies signature validity and disables project execution if the signature became invalid. Project signing provides an additional layer of security.

Execution of unsigned projects (created in old versions of EasyMorph Desktop), or projects with invalid signature is disabled by default, but can be enabled in the space settings.

File/folder picker locations

The "Locations" tab in the space settings allows specifying drives that are shown in the file/folder picker dialog. The dialog is invoked when a task parameter is specified, and the corresponding project parameter has type "File path" or "Folder path".

Note that the locations only specify which drives are shown in the file/folder picker dialog. They do not restrict physical access to disk drives for particular space, and can't be used to restrict access to local and mapped drives for workflows.



Screenshot 11: File/folder picker locations.

Managing user permissions in spaces

EasyMorph Hub employs a role-based permission management system. Roles are defined individually for each space. Each role has a configured set of permissions and can be assigned to one or more users or user groups.

User roles

There can be two types of roles in EasyMorph Hub:

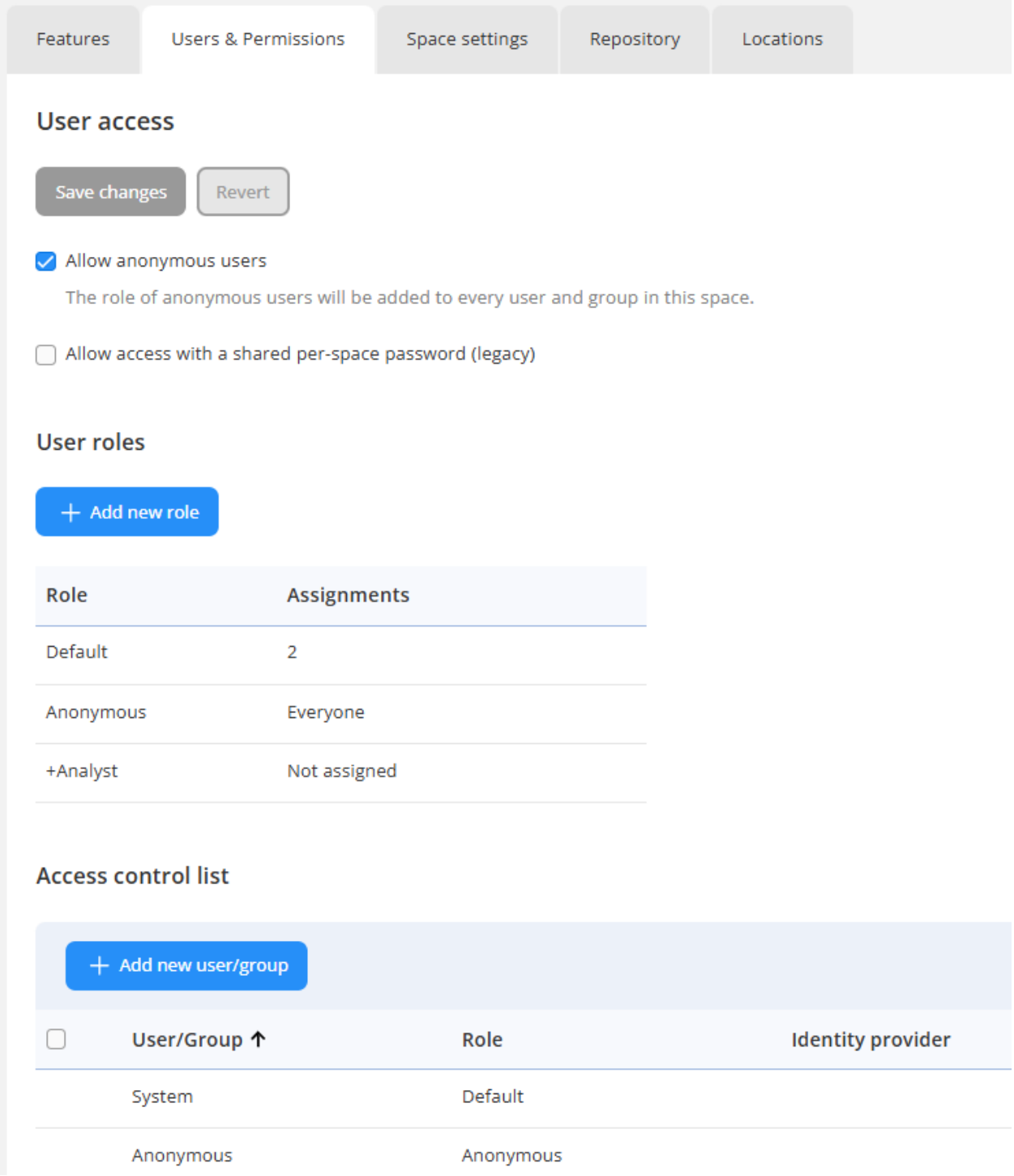
- Regular roles
- Extension roles

Two or more roles (no matter regular or extension roles) can be assigned to the same user (or user group). In this case, their permissions are *combined* (union-ed).

Both types have almost identical sets of permissions and can be combined without limitations. However, they are used differently:

A *regular* role allows accessing the space and is self-sufficient.

An *extension* role by itself can't allow accessing the space even when it's assigned to a user. It is only effective when applied in a combination together with a regular role. Effectively, it *extends* the set of permissions provided by a regular role. You can think of it as of a permission add-on. Extension roles allow more fine-grained control and are marked with the plus (+) sign (see the screenshot below).

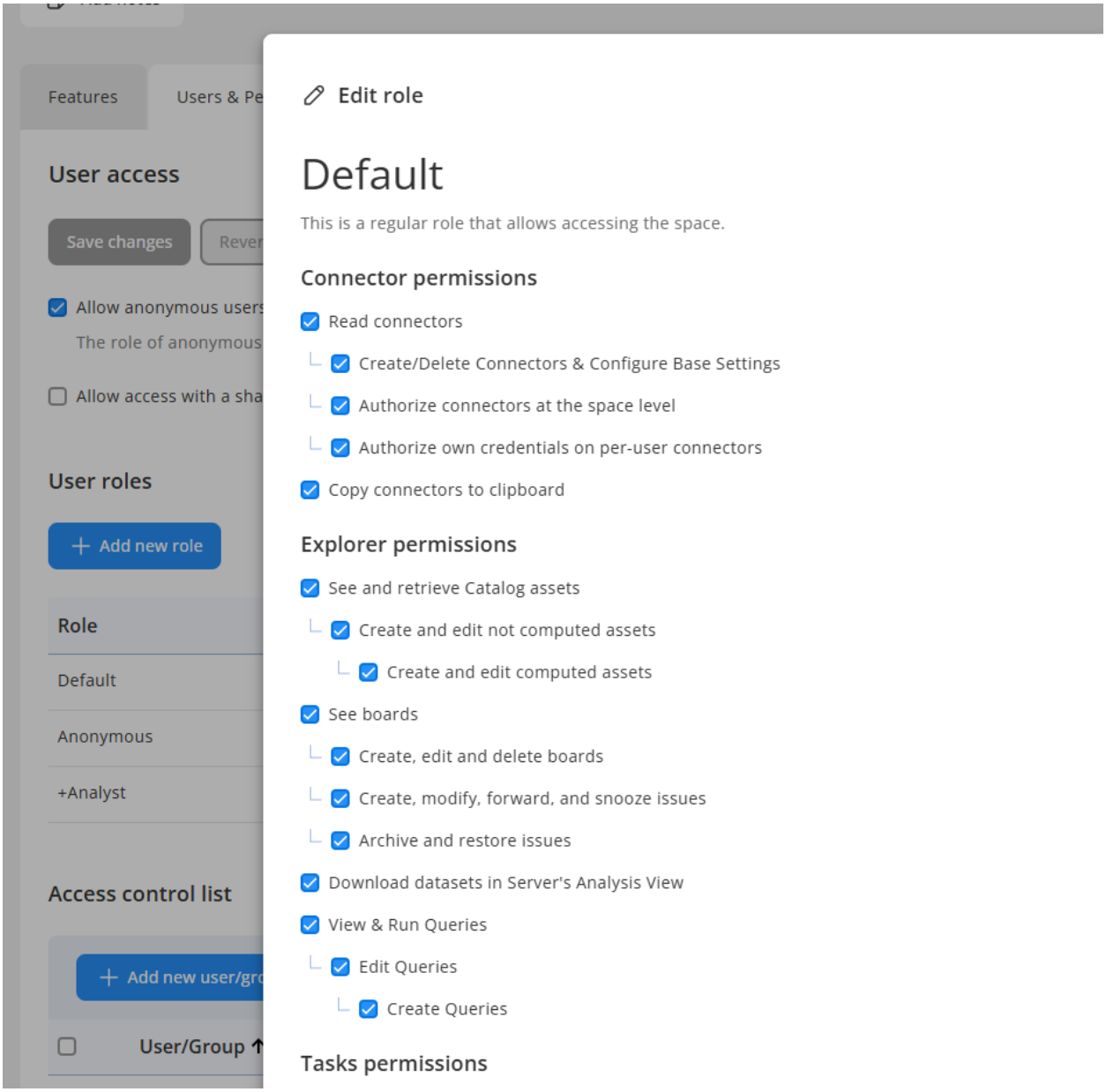


Screenshot 12: User roles of a space.

In a typical use case, basic permissions are provided to users implicitly through an AD group when a role with the basic permissions are assigned to the group. More advanced permissions are provided to some selected users by explicitly assigning them an extension role (or roles) with more advanced permissions. In this case, if one of the selected users is removed from the AD group, s/he automatically loses access to the space no matter if s/he had advanced permissions assigned via the extension role.

Role permissions

Click a role in the list of roles to edit its permissions. The permission configuration dialog is depicted below:



Screenshot 13: Permission set of a role.

Let's break it down:

Repository permissions for Desktop users (only applied in EasyMorph Desktop):

Permission	Description

Read connectors	Desktop users can see and use in workflows connectors from the space's connector repository.
Create, edit, and delete connectors & configure base settings	Desktop users can create, authorize, edit, and delete connectors. For "per-user" connectors, Desktop users can configure the base settings (that exclude authorization credentials).
Authorize shared connectors	Authorize connectors with credentials that are shared with all space users.
Authorize per-user connectors with own credentials	Authorize "per-user" connectors using own credentials. These connectors can only be used by the user who did the authorization. If other users need to use these connectors, they must authorize them with their own credentials.
Copy connectors to clipboard	Desktop users can copy connectors to clipboard (e.g. to paste in another space, or in an own file-based repository, or embed in workflows). Note that this operation allows copying and using somewhere else a connector without knowing the credentials stored in the connector, which can be a security risk.

Explorer is explained in the dedicated section "Explorer" further in the guide.

Explorer permissions:

Permission	Description
See and retrieve Catalog assets	See and use Catalog assets directly or from Explorer boards.
Create and edit not computed assets	Create, modify, and delete assets that are not computed (e.g., with a workflow).
Create and edit computed assets	Create, modify, and delete assets that are computed (e.g. using a workflow).
See boards	See a list of Explorer boards and their contents.
Create, edit, and delete boards	Create, edit, and delete boards. Note that deleting a board also archives all issues in that board.
Create, modify, forward, and snooze issues	Create, modify, forward, and snooze issues. The "New issue" button isn't available if this permission isn't granted.
Archive and restore issues	Archive and restore issues. Note that seeing archived issues is still possible even if this permission isn't granted.
Download datasets in Hub's Analysis View	This permission enables/disables the "Download" button in the Analysis View. Note that this permission doesn't prohibit downloading datasets in EasyMorph Desktop or retrieving them using a workflow.
View & run queries	Open querybooks and run queries in them.
Modify queries and create new queries (without saving them)	Modify queries, add new queries, but not save querybooks. All modifications are lost when the browser tab with the querybook is closed.
Create, edit, delete, and save querybooks	Create querybooks, edit the querybook name and connector. Save changes in querybooks.

Hub Tasks permissions:

Permission	Description
See and run tasks	See and run existing Hub tasks. Change task parameters (if prompted for).

Edit task settings (except project path)	Edit any setting of a task, except the project path. The project is locked meaning that the project path can't be changed.
Create new tasks and change project path	Create new tasks and change the project path in the settings of existing tasks.

Note that tasks that are scheduled or started by a trigger are always executed under the special System user (see "Special user accounts" below) that also has an assigned role.

File Manager permissions:

Permission	Description
Download files	Download files via File Manager.
Upload and delete files	Upload and delete files via File Manager.
Access remote locations	Access remote locations such as SFTP or AWS S3 via connector.
See, hide, and unhide hidden folders	See, browse, hide, and unhide folders. A hidden folder is a folder that have the "Hidden" attribute set.

Note that these permissions do not restrict physical access to the public folder. They only restrict the functionality of File Manager provided to the user. If a Windows account has full access to the public folder, setting File Manager permissions to "Download files" only won't prevent other applications or scripts that use that account from accessing files in the public folder.

Also note that when accessing remote locations, access permissions for those locations are defined by the credentials/accounts configured in respective connectors.

SFTP server permissions:

Permission	Description
List files and folders	View the contents of a folder exposed by SFTP.
Download files	Download files.
Upload and delete files	Upload, rename, and delete files.

Web Automation permissions:

Permission	Description
Add, edit, and delete external users	Add, edit, and delete user accounts in web spaces (don't confuse with Hub spaces) configured in the Web Automation feature.
View, add, edit, and delete web tasks	View, create, edit, and delete web tasks.

By combining different roles and permissions, you can flexibly configure spaces for different types of users and use cases. For instance:

Use case	Task permissions	File Manager permissions
External data suppliers that are required to only provide files with source data.	Feature "Tasks" is disabled	Upload and delete files
Marketing analysts that upload files, trigger pre-configured tasks, and collect results.	See and run tasks	Full access

Sales department employees that need to generate a report on demand. They run a pre-configured task and provide their employee ID as a task parameter.	See and run tasks, Edit task settings (except project path)	Download only
Power users that use a dedicated server to perform ad hoc heavy data transformations.	Full access	Full access

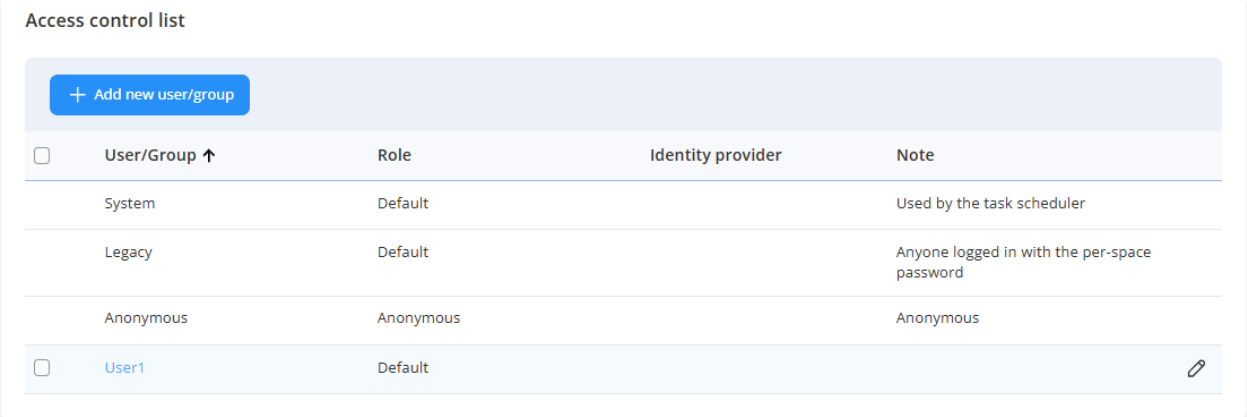
Table 1: Examples of use cases for spaces.

Access control list

The access control list (ACL) defines who can access the space and its user content in various space features such as tasks, files, or Catalog assets.

If the Active Directory (further, AD) connection is configured in the Hub settings, AD users and groups can be added right in the ACL without prior creation in the "Users" tab. The necessary Hub user accounts and user groups will be created automatically.

Each ACL record assigns one or several roles to a user or a user group. If several roles are assigned, their permissions are combined.



Screenshot 14: Access control list of a space.

Note that effective permissions can combined implicitly via group membership. For instance, a user has a role explicitly assigned to him/her in the ACL. However, if the user is a member of an Active Directory group that also has a role assigned in the same ACL, then s/he will get implicitly the permissions the group's role as well.

Due to the security-critical function of the ACL, the system file with ACL records must never be edited outside of EasyMorph. The file and other space configuration files are digitally signed and will be rejected by the Hub if tampered with. As a result, access will be blocked for all space users.

Special user accounts/roles

Besides the regular user accounts, there are three special built-in user accounts in EasyMorph Hub:

- System user

- Anonymous user
- Legacy user

The "System" user performs autonomous actions of EasyMorph Hub, such as triggering a workflow on schedule. Any role or a combination of roles can be assigned to this special user.

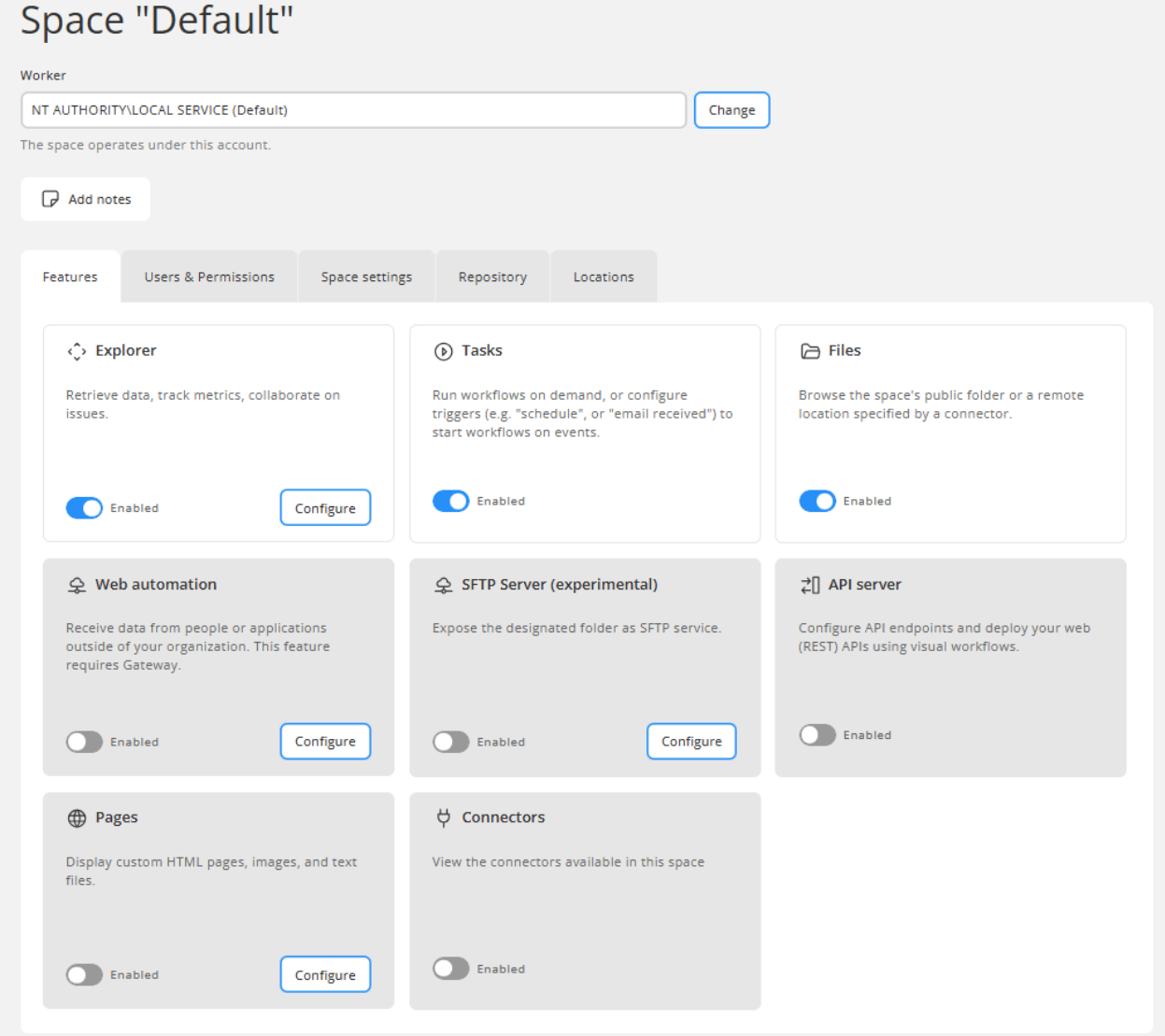
The "Anonymous" user represents anyone who is accessing the Hub anonymously. This user can only have the special "Anonymous" role which is also automatically added to everyone in the ACL of the space (i.e. combined with whatever role(s) everyone is assigned). The "Anonymous" user and the "Anonymous" role are only available when anonymous access is allowed in the space settings.

The "Legacy" user represents the account-less access with a shared per-space password used in EasyMorph Hub versions before 5.8. Any role or a combination of roles can be assigned to this special user.

Using the "Legacy" user for accessing spaces is not recommended for purposes other than supporting compatibility with previous versions.

Space features

The features (functionality) available to the users of each space can be configured individually in the space settings. Each feature can be turned on/off individually. Some features may require additional configuration.



Screenshot 15: Space features.

Explorer

Explorer allows retrieving data from multiple data sources and exploring it, creating collaboration boards with metrics and charts, and managing team work using highly automated issues (tickets).

Explorer is an extensive feature and provides a lot of functionality. It is explained in more detail in the dedicated chapter "Explorer" further in this guide.

Tasks

Hub tasks is the most frequently used feature of the Hub. While Explorer is intended for the "human-to-machine" type of interaction with the Hub, Hub tasks are well suited for "machine-to-machine" automation scenarios and unattended use.

A Hub task runs an EasyMorph project (workflow) and can be triggered manually by any user who is authorized to access it and who has an assigned user role with that permits running tasks.

Besides, that a task can be started automatically by one of the following triggers:

- Scheduler
- Email received
- Database rows added
- File event (file appears/changed/deleted in the specified Hub folder)
    - **Important!** File events are not guaranteed to work shared network folders.
- Dropbox folder changed (file appears/changed in the specified Dropbox folder). This trigger requires a configured connection to EasyMorph Gateway.
- SharePoint folder changed (file appears/changed in the specified SharePoint folder). This trigger requires a configured connection to EasyMorph Gateway.
- SharePoint List changed (a row created/deleted/updated in the specified SharePoint list). This trigger requires a configured connection to EasyMorph Gateway.
- Web response changed

Besides that, there are a few ways to trigger a task from an external application:

- Reload webhook – every task has a unique URL by opening which one can trigger the task.
- Gateway webhook – external (cloud) webhooks that are configured on EasyMorph Gateway trigger the specified Hub task when a new webhook request (message) is received on the Gateway.
- Hub CLI – a cross-platform command-line utility for triggering Hub operations remotely.

Tasks can be configured to prompt for workflow parameters when started by a user. Every task run is recorded in the Hub journal.

When a task fails, the Hub can automatically send an email notification to the specified email recipient. A working email connection must be configured in the "Notifications" tab.

API Endpoints

With the "API Endpoints" feature, EasyMorph Hub works as a custom web API (HTTP) server that you can use to deploy custom internal APIs for real-time machine-to-machine automation, or for retrieving data directly from EasyMorph in applications such as Excel, Power BI, Tableau, Qlik, or any other application that supports the OData protocol (v4).

The API server endpoints can be of two types:

- REST endpoints (incoming requests are served with responses produced by a workflow)
- OData endpoints (incoming requests are served with data from an EasyMorph dataset .dset file)

To create a REST endpoint:

1. Create an API endpoint in the “REST endpoints” tab.
2. Specify a project that should process incoming HTTP requests
3. Assign modules of the project to HTTP verbs
4. Assign module parameters with request properties (e.g. request headers)

Notes on REST endpoints:

- To learn more about designing API endpoints, read this article: "How to create API endpoints without coding".
- You can assign various modules of the same project to respond to different HTTP verbs. For instance, one module can respond to GET requests, while another can respond to POST requests.
- The response latency is about 20-30 milliseconds which makes it possible to serve a million of requests or more per day.
- Every workflow run is recorded in the Hub journal. Therefore, if your Hub processes a large number of API calls, make sure you manage and control the journal database size by deleting (archiving) old records. See also "Journal cleanup".

To create an OData endpoint:

1. Create an API endpoint in the “OData endpoints” tab.
2. Specify a .dset file that contains data returned in response to incoming requests.
3. Configure how the key property is defined (auto-calculated column, or existing column).
4. Configure a list of entity properties (dataset columns) and their data types. The endpoint will only return column specified in the list.

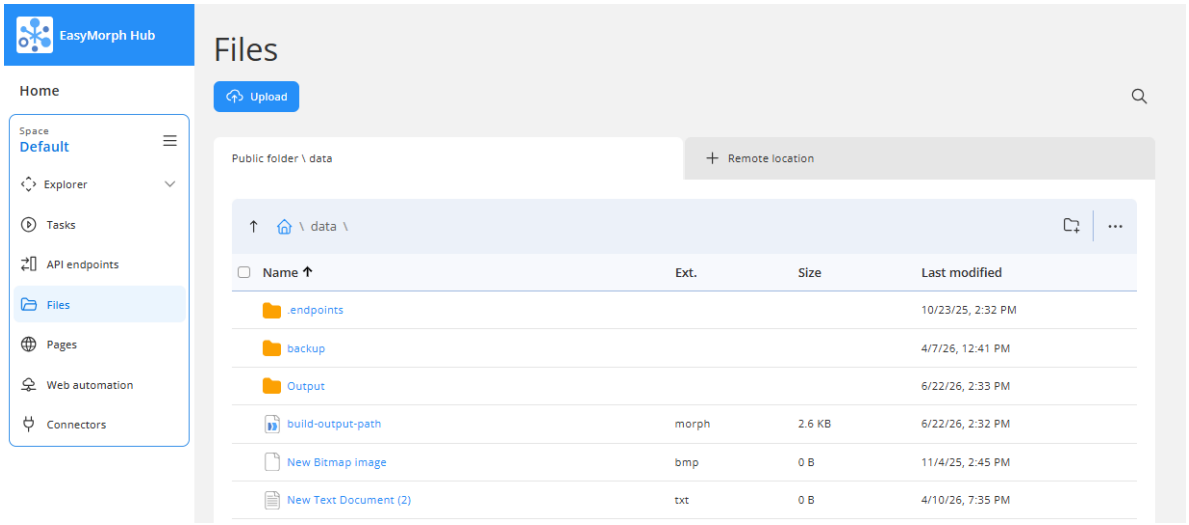
Notes on OData endpoints:

- OData endpoints respond only to GET requests.

The API server by default doesn’t have a configured authentication. It can be configured in the API server settings.

File Manager

The tab "Files" opens the File Manager. It allows browsing the public folder of a space as well as remote locations such as Google Drive or SFTP. To work with a remote location, a connector must be selected.



Screenshot 16: File Manager.

The File Manager allows performing the following file operations:

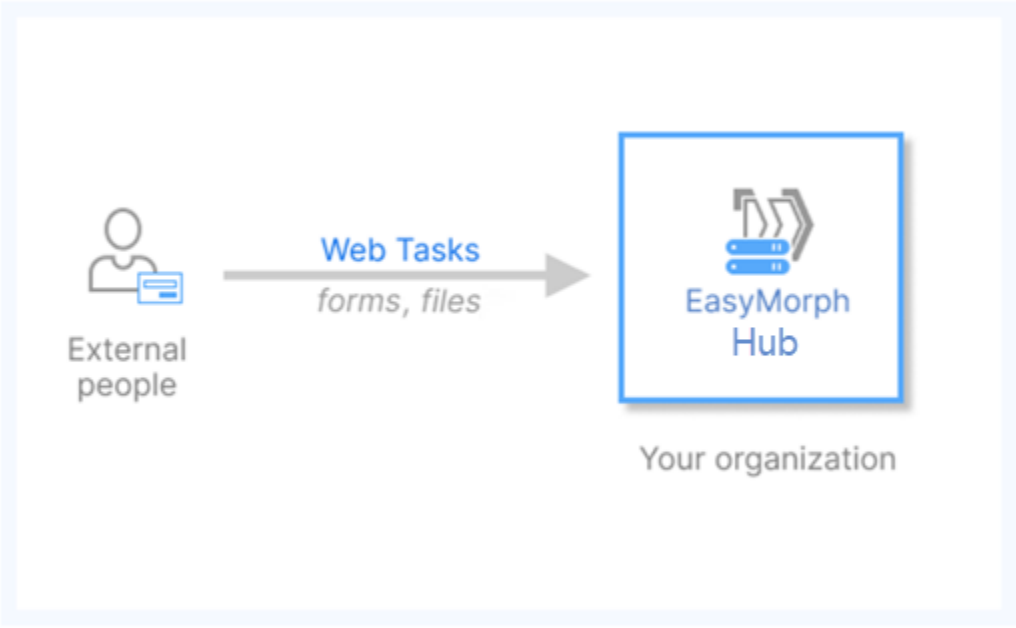
- Browse folders
- Upload one or multiple files by dragging them into browser, or by pressing the “Upload file” button
- Download a file by clicking its name
- Create, rename, or delete a folder
- Rename the selected file
- Delete one or more selected file(s)
- Find a file in the current folder or in subfolders, recursively

Some of the operations above are not possible in remote locations.

Note that in the current version it’s not possible to delete Hub folders that are not empty. All files and folders must be removed from a folder and only then it can be deleted.

Web automation

EasyMorph Gateway (described further in this document) provides a web portal that allows people outside of your organization to trigger Hub workflows and enter workflow parameters. This feature is called “Web automation”. Your organization’s partners, suppliers, customers, and subcontractors can use “Web automation” to submit various data and files to your organization, or trigger various automation scenarios (e.g. request and receive datasets or documents by email, or initiate data upload when it’s ready).



To distinguish such external users from Hub users, they are called *web users.*

Web user can access a *web space* – a section of the portal that is connected to a Hub space. A web space can be configured to use an authentication mechanism (currently, it's email or Microsoft account). Also, each web space has a list of web users that can access it (so called ACL, access control list).

Finally, there is a separate mechanism of *web tasks* – Hub workflows that can be triggered by web users via Gateway's web portal. So, web users log into the portal, pick a web task, and run it.

Web tasks, web spaces, and web users are configured in the tab "Web automation" of EasyMorph Hub.

To use "Web automation", the Hub must have a configured connection to EasyMorph Gateway.

SFTP Server

EasyMorph Hub comes with a built-in SFTP server that can be enabled in each space individually. The SFTP service allows transferring files to and from EasyMorph Hub using any standard SFTP utility.

The general SFTP settings are configured in the general Hub settings (tab "Settings"). Note that SFTP service runs a separate port, also configured in the settings.

In each space, when the "SFTP Server" feature is enabled, it exposes the specified subfolder of the Public folder (or the Public folder itself, by default) via the SFTP service.

Different spaces appear as first-level folders on the SFTP service, where each folder corresponds to a space and is named accordingly.

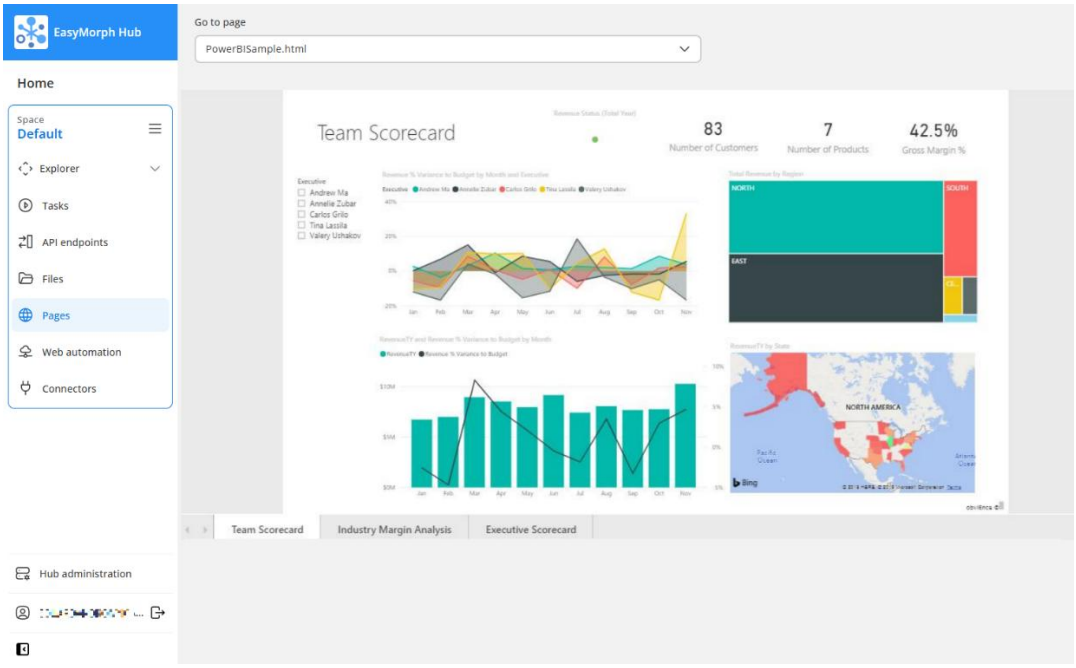
Client authentication for the SFTP service is done using the same username and password as for accessing EasyMorph Hub.

Pages

When enabled, the feature "Pages" offers a simple built-in web-server that can display custom static HTML pages to the users of a space. The HTML files must be located in the Pages' root folder that is specified in the Space settings. The folder can have subfolders with HTML files as well. When a file named **index.html** is present in the root folder, it's displayed by default. Otherwise, Pages display a random HTML file that happened to be found first. HTML pages can display any information that a web-page can display. For instance:

- Welcome page
- An HTML table with result data generated by an EasyMorph task
- 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 Hub usage statistics (RAM, CPU, etc.)

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



Screenshot 17: Custom HTML page in the Pages tab featuring an embedded Power BI dashboard.

Explorer

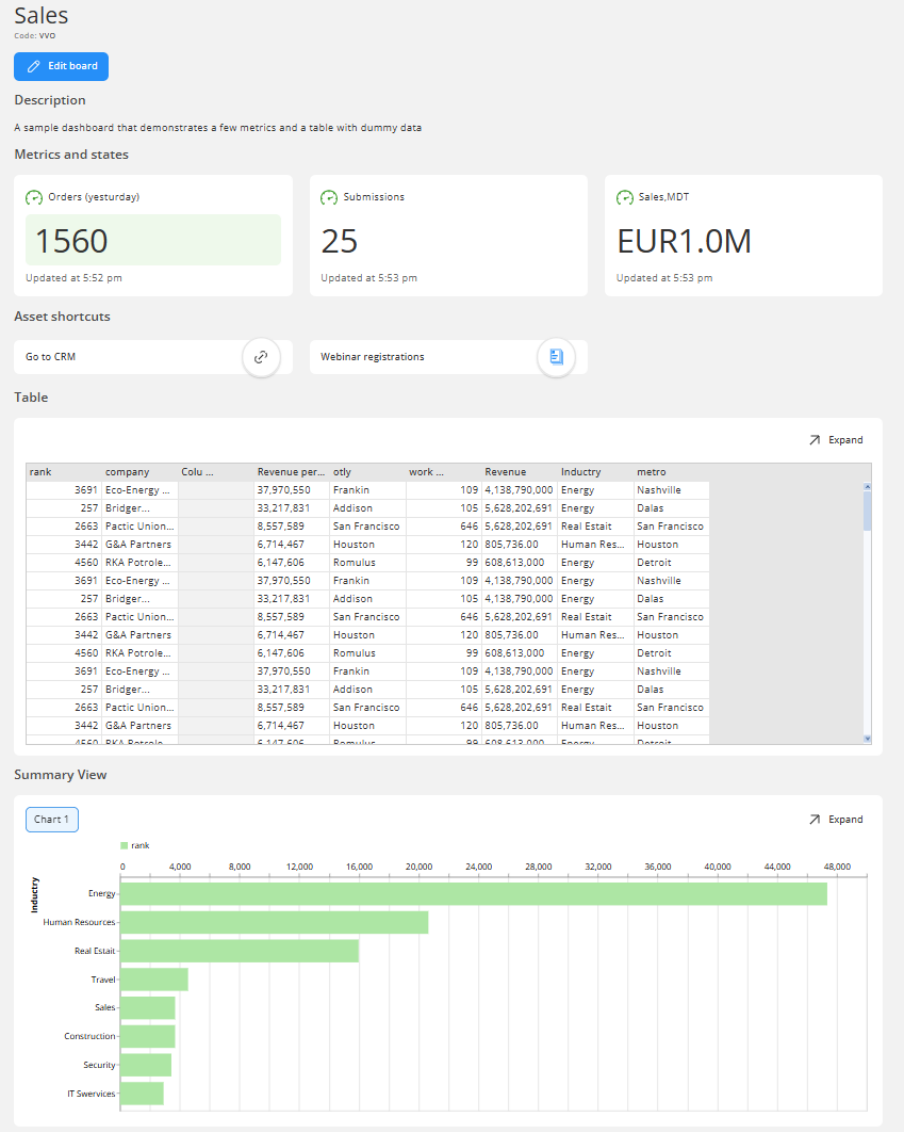
With the help of Explorer, users can do exploratory data analysis, track metrics, automate their work using an issue tracker and contextual workflows, and collaborate on data and files.

Explorer has the following components that work together:

- Boards
- Catalog of data assets
- Issues
- Queries

Boards

Boards are generally used as interactive collaborative homepages for teams, projects, lines of business, business processes, or business domains. They provide a consolidated view and access to the business information, such as metrics, goals, dashboards, BI reports, datasets, file locations and applications that business users work with on a daily basis.



Screenshot 18: A board in Explorer.

Boards are constructed from various *sections*. In turn, sections are composed from Catalog assets described above. The following board sections are available:

- Issues
- Metrics and States
- Table
- Charts
- Multi-line text
- Asset shortcuts
- File Manager (for a local or remote folder)
- Tasks
- Embedded Tableau workbook
- Embedded Power BI report

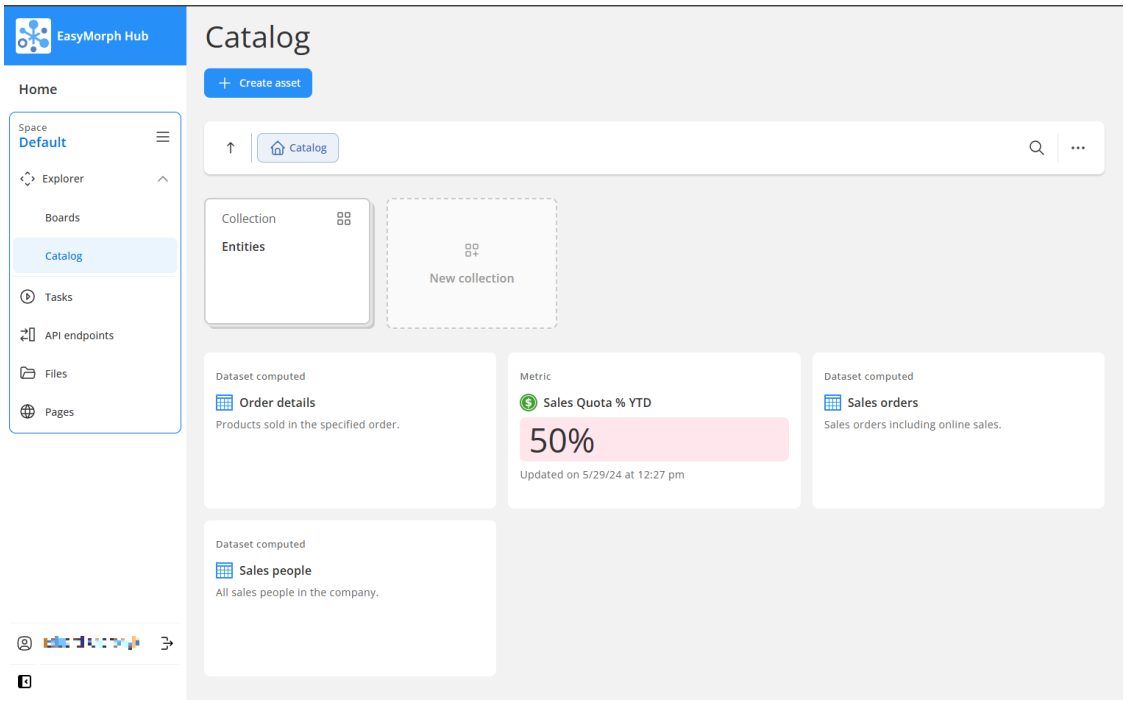
A board can have any number of any sections, except for "Issues". Only one "Issues" section can be present in a board.

Catalog

The Catalog is a hierarchical directory of various data assets that business users frequently use or refer to. The assets can be organized into nested collections. The following asset types can be added to the Catalog:

- Metrics (numeric KPIs) backed by SharedMemory
- States, text labels backed by SharedMemory
- Datasets (tabular datasets stored in .dset files)
- Computed datasets (virtualized tables computed on demand with workflows)
- Visual database queries, database tables and views
- Files (e.g. spreadsheets or PDF documents)
- Folders (including folders in remote locations such as SFTP or SharePoint)
- Automation workflows
- Web resources (URLs), including those computed on the fly with workflows
- BI resources such as Power BI and Tableau dashboards

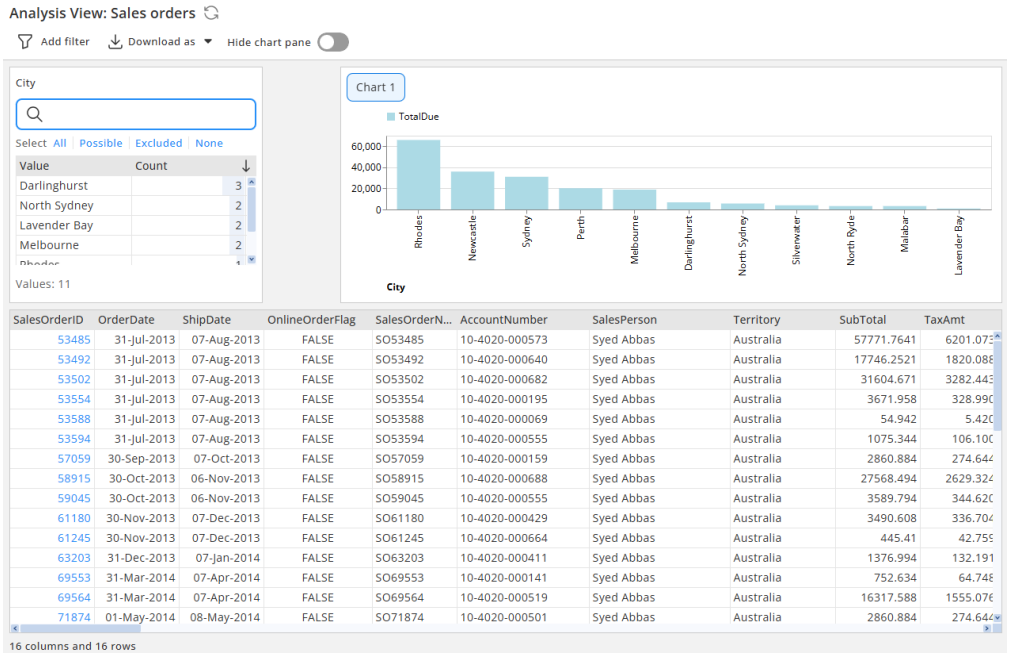
The catalog is searchable, so users can search assets by name, description, or by field name, or field description.



Screenshot 19: Catalog assets.

Catalog assets can be opened or retrieved directly from the Catalog, or they can be used as building blocks for boards or as attachments in issues.

Datasets can be opened, viewed, and filtered right in EasyMorph Hub's web UI. They can be produced on the fly using EasyMorph workflows which parameters can be entered before dataset retrieval.



Screenshot 20: Analysis View of a computed dataset.

Besides that, the datasets can be *hyperlinked* – when clicking a cell value in one dataset opens a *computed* dataset and the cell value is assigned to a workflow parameter of the computed dataset. This allows creating and exploring a virtual "web of data" – a set of interconnected dynamically computed datasets with data from various business applications needed for business users. In a way, it can be viewed as a virtual data warehouse for business users.

Hyperlinking datasets is done by configuring asset commands in the settings of dataset assets (computed and pre-computed).

Issues

Every board can have a built-in issue tracker. The issue trackers are intended for *work automation*. In combination with Explorer boards and Catalog assets, described above, issues enable a wide range of business automations. For instance:

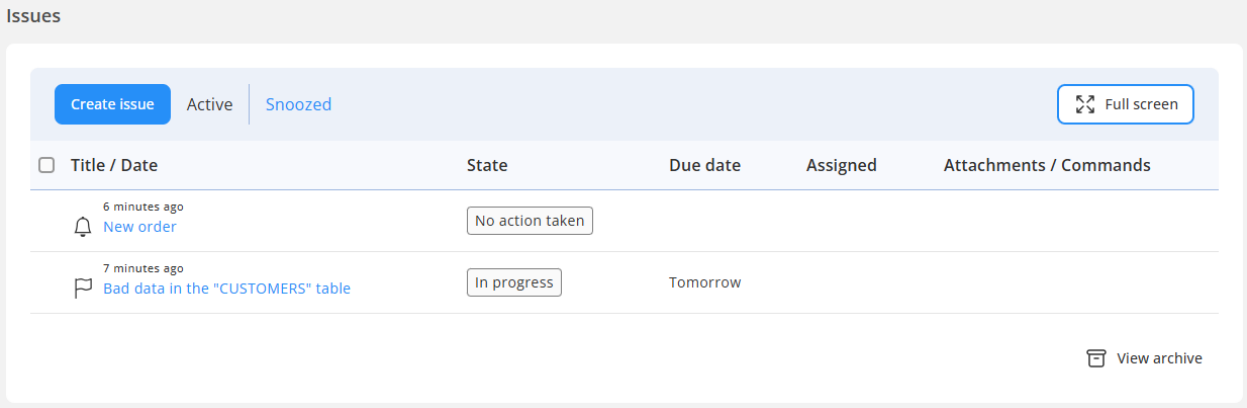
- Detect and collaborate on data quality problems
- Generate alerts about business metrics and system health that can be converted into tasks assigned to designated users
- Receive and collaborate on support tickets from internal and external customers
- Reconcile incoming payments with insufficient details
- Build processing pipelines for customer orders placed online or for job applications
- Detect, alert about, and collaborate on suspicious transactions
- Plan and coordinate team work on workflows
- Provide analytical remarks and context for important KPIs

What is an issue? There are multiple ways how one can think of issues. For instance, you can think of them as of an advanced to-do list where each to-do item can be assigned to a person and have a deadline. Or,

issues can be viewed as something akin to email messages, where each message has a subject, a text, and various attachments, and can be forwarded from one inbox to another inbox.

Issues can be of different types (e.g. "alert", "request") and can be in different states (e.g., "received", "in progress", "delayed"). Issue types and states are configured in the space settings.

Issues are available as a new section type in Explorer boards. For now, one board can have only one section with a list of issues.



Screenshot 21: Section "Issues".

Issues can be created in two ways: manually and programmatically. To create an issue manually, add a section "Issues" to a board, and press the "New issue" button.

Alternatively, you can create issues programmatically in a workflow, using the "Raise issue" action.

Issue *attachments* is arguably the most interesting feature of the issue system. The attachments can be of different types:

- Files and folders in the public folder of the space or folders in remote locations (such as Google Drive or Amazon S3)
- Web links
- Hub tasks with pre-configured parameters
- Catalog assets (parameters of computed assets can be pre-configured)

The latter, Catalog assets, assumes any asset type, including dynamically computed assets such as computed datasets and computed web links, pre-filtered Tableau and Power BI reports, or workflows. When creating an issue (especially, programmatically), you can attach a computed dataset to the issue, and pre-configure parameters of the asset. An attachment can be opened with a single button click right from the list of issues. For instance, when an issue warns about a data quality problem in a database table, it can have an attachment that in one click immediately opens in Explorer the subset of table rows in question.

Or, if an issue is created for an incoming payment that need to be matched with an invoice (i.e. reconciled), with one click you can open an attached computed dataset that contains all the open invoices with the matching invoice amount.

In a similar fashion, you can attach *workflows* to issues, for instance to approve or reject requests, push data (as pre-configured workflow parameters) into a web API, or continue processing orders in another system.

Finally, issues that have been processed or are no longer necessary can be *archived*. Note that issues are created permanently and can't be deleted, only archived. Therefore, be careful with creating a big number of issues programmatically.

To learn more about issues, see this Community article: "Getting started with work automation in EasyMorph".

Advanced Hub configuration

System folder

The system folder is configured in the "EasyMorph Hub Monitor" utility. It contains the internal Hub files required for Hub operation. Note that the system folder doesn't contain user workflows or data – they are located in the public folder specified in the space settings. It is recommended to not change the path of the System folder. However, if you need the system folder to be in another location, make sure you use a local drive, not a network drive.

Using network drives (mapped or not) for the system folder is **strongly not recommended** because network drives don't provide the necessary file consistency guarantee[1] for the SQLite database that is used internally for storing Hub configuration.

Note that the System folder and all the files and subfolders in it must permit full access for the "EasyMorph Server" Windows account group. The group is created by the EasyMorph installer automatically.

The "EasyMorph Server" group

The Hub installer automatically creates the Windows account group "EasyMorph Server". The group contains[2] the Service account. The group simplifies folder access management, for instance, when the Server service account needs to be changed. By default, the system folder, the log folder, and the Public folder of the Default space also allow full access to the group "EasyMorph Server". If you need the Server service account to access some other folder, allow the "EasyMorph Server" group to that folder, instead of allowing the Service account explicitly.

Installing EasyMorph Hub on a machine that serves as Windows Domain Controller is not recommended, but still possible. In this case, the Hub installer will fail to add the Server service account (explained below) to the group due to Windows-specific restrictions on Domain Controllers. You will have to

[1] See https://www.sqlite.org/draft/useovernet.html.
[2] The "EasyMorph Server" group was introduced in v5.2 and can be empty if your Hub was installed from an earlier version. In this case, just add the current Service account to this group.

manually create or ensure that the access group "EasyMorph Server" (the exact name and case matters) exists already. Create a technical user account to run EM Server service, set the Server service to use this account as Logon account in Windows service settings, and finally, add this technical account to the "EasyMorph Server" group. Failing to do any of these steps could lead to hard-to-diagnose issues (e.g., the Server service having no/partial access to its own configuration files).

Server service account

By default, the Server service is installed under the local Windows account *NT AUTHORITY\LocalService*, not the account under which the installer was run. The *NT AUTHORITY\LocalService* account is a standard account for Windows services. It has fewer privileges than an administrator account. For instance, it can't access the users' Documents folders and other protected locations. If you need the Hub to access a particular folder, there are 3 possible solutions:

- make sure that the service account has been given necessary access permissions;
- or, switch the Server service to using a different Windows account that has the necessary permissions;
- or, configure a Hub worker to use a Windows account that has necessary permissions in the Hub settings, and use the worker for the space which tasks require the permissions. Read more about Hub spaces and workers in chapters "Spaces and their configuration" and "Workers", respectively.

Also, when using "Windows authentication" in the connector properties, keep in mind that the connection will be established using the Windows account specified in the settings of the Hub space worker from which the connection is established. Use explicit login/passwords for accessing databases, instead of integrated Windows authentication, if necessary. Alternatively, add the Default worker's Windows account (by default it's *NT AUTHORITY\LocalService*) to database logins.

Changing Server service account

Generally, it's not recommended to switch the Server service account after the Hub was in use. In the Enterprise edition it is possible to configure spaces to use different Windows accounts (see chapter "Workers"). Consider using this capability before changing the service account.

If you still need to switch the Server service to another Windows account:

1) Add the new account to the "EasyMorph Server" Windows account group.
2) Make sure that the new account can assess the public folder of the Default space or other spaces where the Default worker is used.
3) Stop the service.
4) Change the "Log On" account setting in the EasyMorph Server service properties in Windows Services panel to use the new account.
5) Start the Service.
6) Remove the old account from the "EasyMorph Server group".

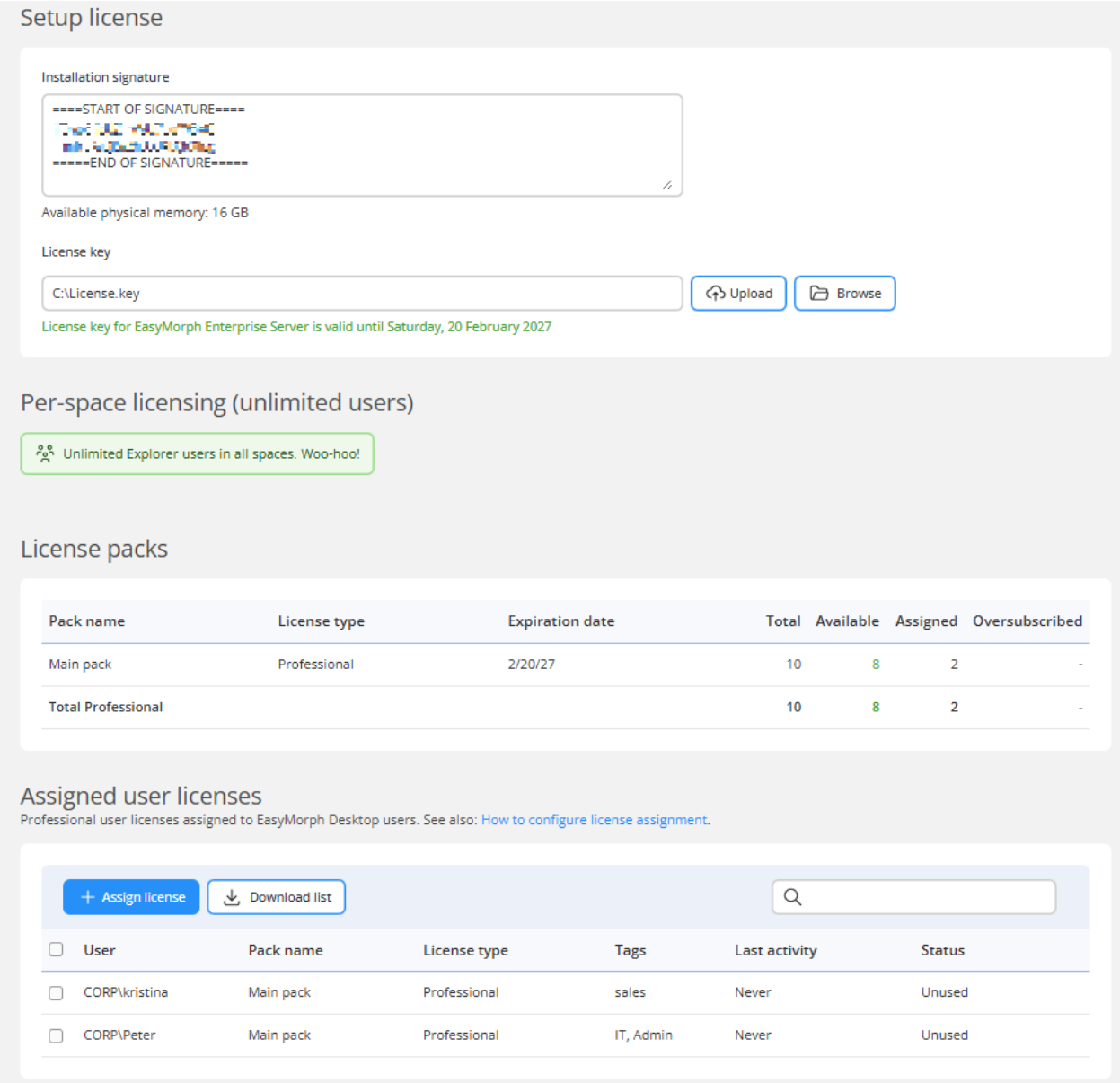
Managing Desktop user licenses

In EasyMorph deployments without EasyMorph Hub, Desktop users are provided with a "static" license encoded directly in a license key file. For easier user license management, EasyMorph Hub supports "dynamic" licenses that can be assigned or transferred to Desktop users right from the Hub by a Hub administrator. A "dynamic" license is a regular Professional user license for EasyMorph Desktop.

Assigning Dynamic licenses

A license key may include one or more packs of Dynamic licenses. When a license key with at least one valid pack of Dynamic licenses is applied on EasyMorph Hub, the Hub can assign (lease) licenses to Desktop users dynamically.

For example, a pack contains 5 Dynamic licenses. It means that 5 Desktop users can be assigned a license from Hub by a Hub administrator. If a Dynamic license needs to be transferred to another user or revoked, a Hub administrator can do this right from the Hub's web console.



Screenshot 22: Dynamic license assignment.

A new license assignment can be created by pressing the "Assign license" button visible in the screenshot above. When a new assignment is created, it's added to the license assignment table also visible in the screenshot above.

For Desktop users, in order to lease a license from Hub, the Hub Link must be configured (see "**Error! Reference source not found.**" below), and the "Lease from EasyMorph Hub" option must be selected in the "License setup" dialog (invoked by pressing the "Setup license key" in the Home screen of EasyMorph Desktop).

Each license lease lasts for 48 hours and is renewed automatically every 48 hours for each user who is assigned a license from Hub. For a license to be renewed automatically, the Hub Link must remain configured on Desktop, and Hub must be reachable by network.

A license assignment can have one of the following statuses:

- **Unused** – user never requested a license from Desktop.
- **In use** (version number) – user has successfully leased and is currently using a dynamic license.
- **Quarantined** – license was recently deleted or transferred to another user and therefore is temporarily blocked from use (more on the quarantine below).
- **Oversubscribed** – user should have been assigned a license from a pack but there are more license assignments than available licenses in the pack.
- **Missing pack** – user should have been assigned a license from a pack but the pack is missing in the current license key.
- **Expired pack** – user should have been assigned a license from a pack but the pack is expired in the current license key.

Quarantine

When a license assignment record was deleted or edited, the license that was assigned may be temporarily quarantined. The purpose of the quarantine is to prevent a simultaneous use of a Dynamic license by two or more users (which would be a violation of the software licensing terms). The duration of the quarantine depends on when the license was last used before it was deleted or re-assigned. If the license was used more than 48 hours ago, then no quarantine enforced at all and the license is immediately released. If it was used less than 48 hours ago, then it's quarantined until 48 hours since its last use. Quarantine can never last longer than 48 hours. Once the quarantine ends the license returns to the pool of available licenses and can be assigned to another user.

Oversubscription

If the number of Dynamic licenses available in a license pack is less than the number of users that should be assigned a license from the pack, some license assignments will not be successful and will switch in the "Oversubscribed" status. For instance, if pack "Marketing" contains 5 licenses, but the assignment table has 12 users assigned a license from this pack, then 7 assignments (i.e. 12 – 5 = 7) will fail and the respective users won't be able to lease a license from Hub.

In case of oversubscription, the license assignment algorithm favors more active users (i.e. users that most recently used a license). In the example above, the 5 most active users will still be able to lease a license while the 7 least active users won't be assigned a license.

Hub journal

EasyMorph Hub writes a journal that records various user actions and system events, such as:

- Workflow completions (successful or not). The event record includes parameters, errors, notifications, status messages, start and elapsed times, initiator, and other metadata.
- Successful user actions, including
    - Logged in/out
    - Task triggered / changed / deleted

- File uploaded / downloaded / deleted
- Folder created / renamed / deleted
- Repository connector created / edited / deleted
- Catalog item retrieved / created / edited /deleted
- Catalog collection created / edited /deleted

From a technical standpoint, the journal is a database table. Two types of database connections are supported: embedded and ODBC:

Connection	Description	Editions
Embedded (default)	Embedded SQLite database, comes with Hub installation, requires no configuration. No access restrictions. No failover.	All editions
External	ODBC (MS SQL Server or Postgres). Doesn't come with Hub installation and must be configured separately. Allows restricting access. Automatic failover switching to the embedded journal database if the external database is not available.	Enterprise only

Regardless of whether an embedded or external ODBC database connection is used, the journal format is the same.

The embedded journal database is a SQLite file. Its default location is "C:\ProgramData\EasyMorph Server\journal.db".

Failover switching

If the embedded journal database becomes unavailable, the Hub will start accumulating events in memory and keep retrying to access the database. If 500'000 events have been accumulated in memory and the database journal is still not available, the events will be discarded from memory and lost.

If the external journal database becomes unavailable, the Hub will automatically switch to the embedded journal database. Switching back to the external database should be done manually from the Hub's web-console (tab Journal). The records created in the internal database during a failover switch should be copied to the external database manually, if required.

User interface

The journal UI in EasyMorph Hub allows filtering:

- By date
- By project
- By initiator (user, scheduler, API request, etc.)
- By status (success, failed, canceled, etc.)

Filters of the same or different types can be combines, so multiple filters can be applied at once.

Since the journal is just a database table, it can be queried using the Query Editor in EasyMorph, or any SQL-compatible data visualization tool such as Tableau or Power BI.

Besides completed events, the journal displays in real time:

- Currently running workflows (administrators can cancel any workflow in any space, users can cancel workflows in their spaces only)
- Currently logged in users (administrators can log out any user)

This data is not recorded in the journal and is only available via the Hub's web interface.

Task journal

The task journal is available for each task (in task details) and represents a subset of journal records related to task.

Journal cleanup

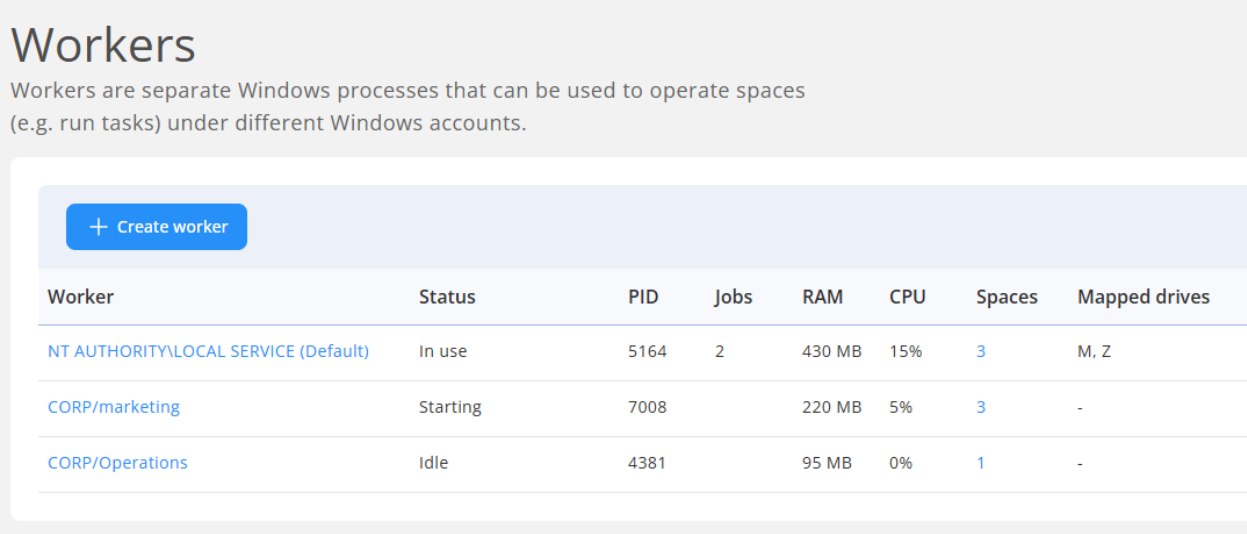
The journal works in the "always append" mode and therefore the number of records in the journal always increases. While the embedded journal database can handle billions of records, over time with extensive Hub use the journal database may grow very large.

EasyMorph Hub doesn't remove old records from the journal database automatically. The journal data retention policy is left to the Hub admins who can design and schedule an EasyMorph project that would automatically remove old journal records.

Workers

A worker in EasyMorph Hub is a Windows process that executes all operations of a Hub space: run tasks, access files, etc. One worker can be used by multiple spaces. Although, one space can use only one worker to perform all operations of the space.

The Server service (with the built-in Default worker) and additional workers are Windows processes with the names **Morph.Server.WebConsole.exe** and **Morph.Server.Worker.exe**, accordingly. Their process IDs (PIDs) can be seen in the "Workers" tab of Hub settings (depicted below). These PIDs correspond to the PIDs that can be seen in Windows Task Manager (tab "Details").



Screenshot 23: Workers of EasyMorph Hub.

The Default worker

The Default worker is a special worker. Unlike other workers, it's embedded in the Hub service and therefore runs under the service's account (by default, it's NT AUTHORITY/LocalService).

In EasyMorph Hub, the Default worker can't be deleted, edited, or recycled.

Note that tasks in spaces that operate under the Default worker have full access to the system folders and files of EasyMorph Hub (such as Hub settings). Therefore, in environments with high demands for security, it is recommended to avoid using the Default worker for spaces at all.

Additional workers (Enterprise edition)

Additional workers are separate (child) processes that are attached to the main process of the Server service. They can run under a different Windows account than the service.

Before a worker can be used for a space, it must be created and configured in the tab "Workers".

Running spaces under different Windows accounts

With the help of workers, EasyMorph Hub allows running tasks and accessing files under multiple Windows accounts. Technically, the Server service does this by spawning and running pre-configured child processes (workers) to execute tasks and access files of designated spaces under the Windows accounts of workers. The picture below shows a sample configuration of workers that allows different spaces to run tasks under different Window accounts. In the configuration below:

- Space 1 and Space 2 use the Default worker (the worker that is built in the EasyMorph Server service)
- Space 3 uses a worker that runs under Account A
- Spaces 4 and 5 use a worker that runs under Account B

- One more worker is configured to use Account C but is not used by any space.

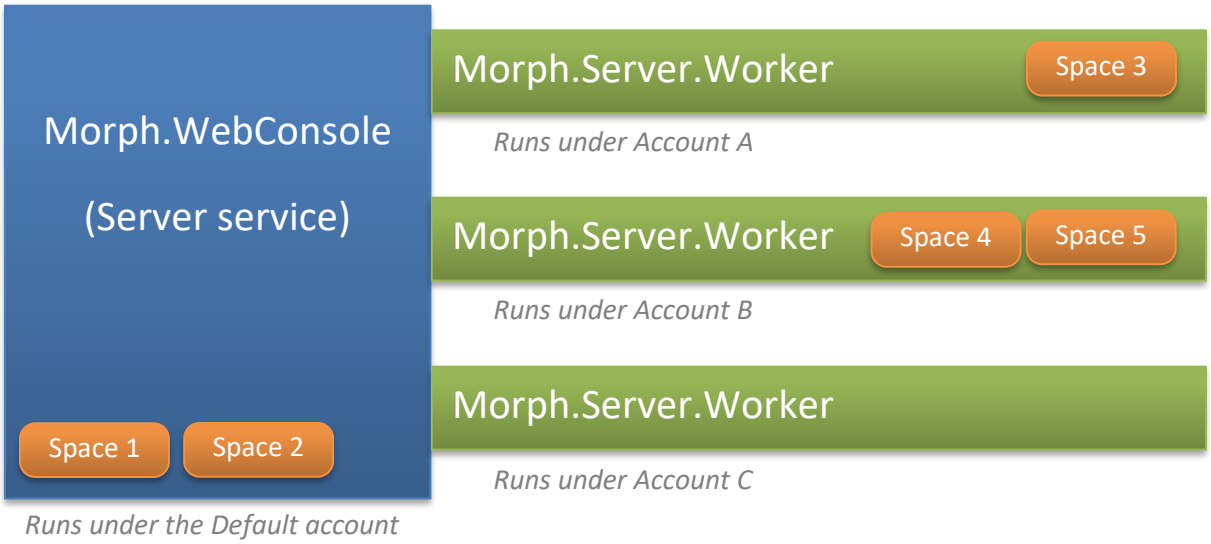


Figure 1: The main (Default) worker and child worker processes of EasyMorph Hub.

Recycling workers

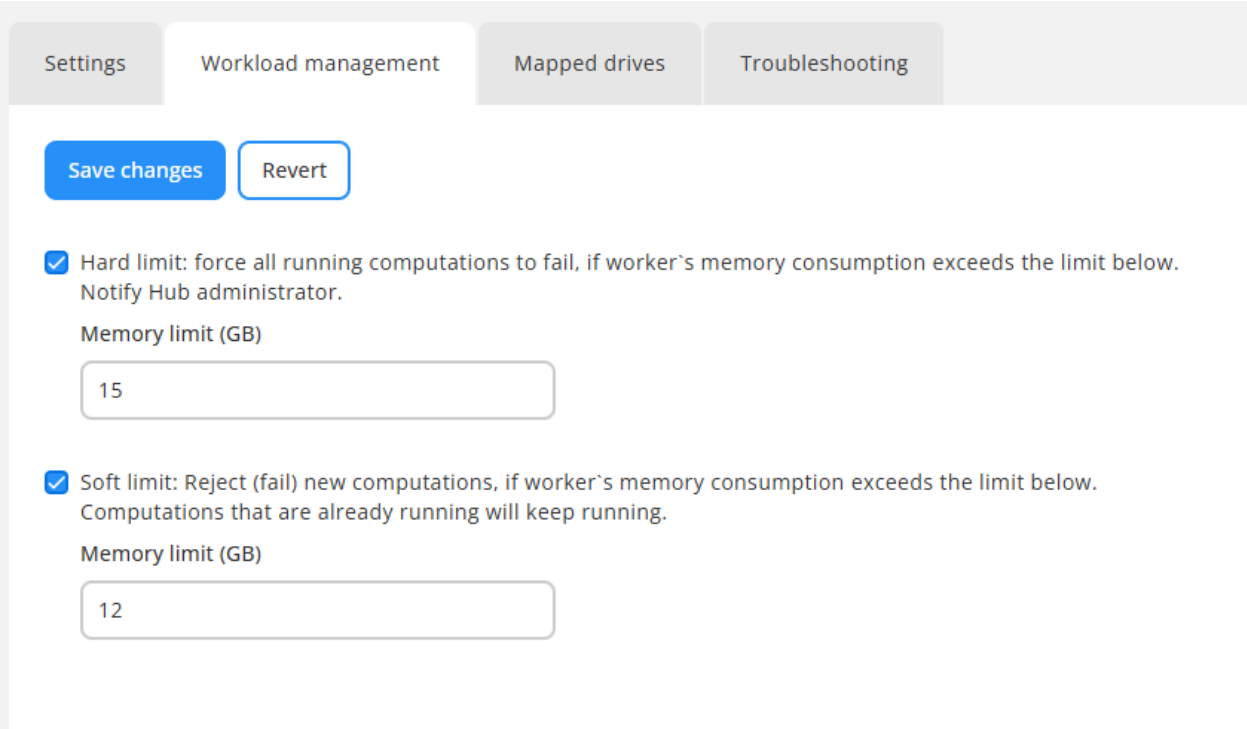
All workers except the Default worker can be *recycled*. Recycling a worker detaches its current process from the Server service and attaches a new one. The detached worker process automatically self-terminates when all tasks that it ran have been finished or cancelled.

Recycling provides a graceful and safe way to forcefully terminate tasks that "misbehave" (e.g. freeze, or lock memory up) as well as delete/reconfigure workers without restarting the Server service. If necessary, remove a detached worker process manually using the Windows Task Manager. The worker process can be identified by its process ID (PID) shown in the list of workers on the tab "Workers" of EasyMorph Hub.

Workload management

Workers can be used for separating and managing Hub's workload in terms of memory utilization. Since the Hub is sensitive to running out of memory, it is desirable to ensure that its memory consumption remains within limits.

You can do it by enforcing "soft" and "hard" memory limits for all or selected Hub workers. The memory limits are configured in the worker settings (not in the space settings).



Screenshot 24: Memory limits in worker settings.

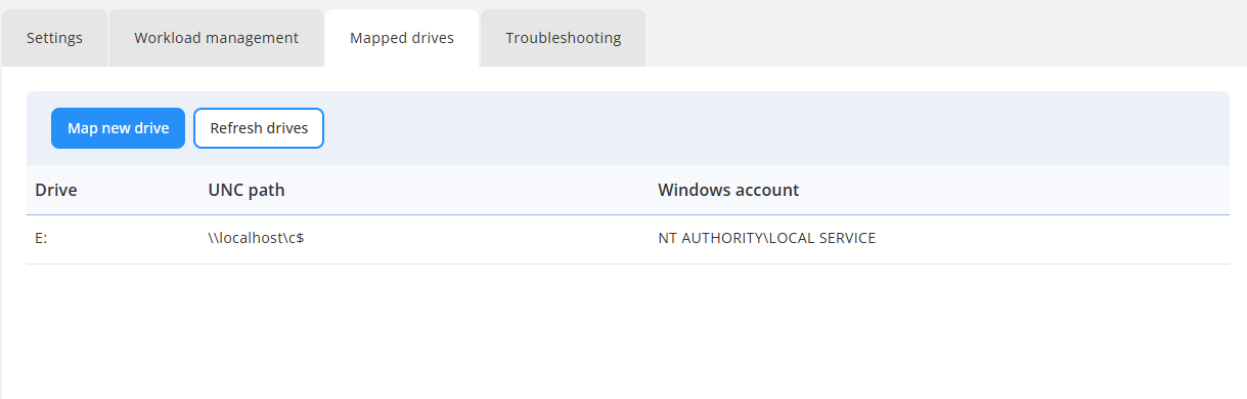
When the "soft" memory limit is reached, the worker will start rejecting all new workflow starts (they will fail) until memory consumption drops below the "soft" memory limit. However, all the currently running workflows will keep running.

When the "hard" memory limit is reached, the worker will immediately cancel absolutely all running workflows.

Note that when the "soft" or "hard" memory limit is configured in a worker, in the "Tasks" tab of all spaces that use the worker the available memory indicator (above the list of tasks) will show the worker's memory limit instead of the total available RAM.

Mapped drives

In Windows, network folders are mapped individually for each Windows account. Therefore, even if a mapped network drive or folder is accessible for one Windows account it may not be accessible by default for another account. In EasyMorph Hub, each worker runs under different Windows accounts. Therefore, to configure mapped drives for a space, open the settings of the worker that is used by that space. Go to the "Mapped drives" tab of the worker (see "Screenshot 25: Mapped drives of a worker." below).



Screenshot 25: Mapped drives of a worker.

To map a new drive, pick an available drive letter, provide a UNC path to the shared network folder, and press the “Map” button. If mapping is successful then the new mapped drive becomes available right away.

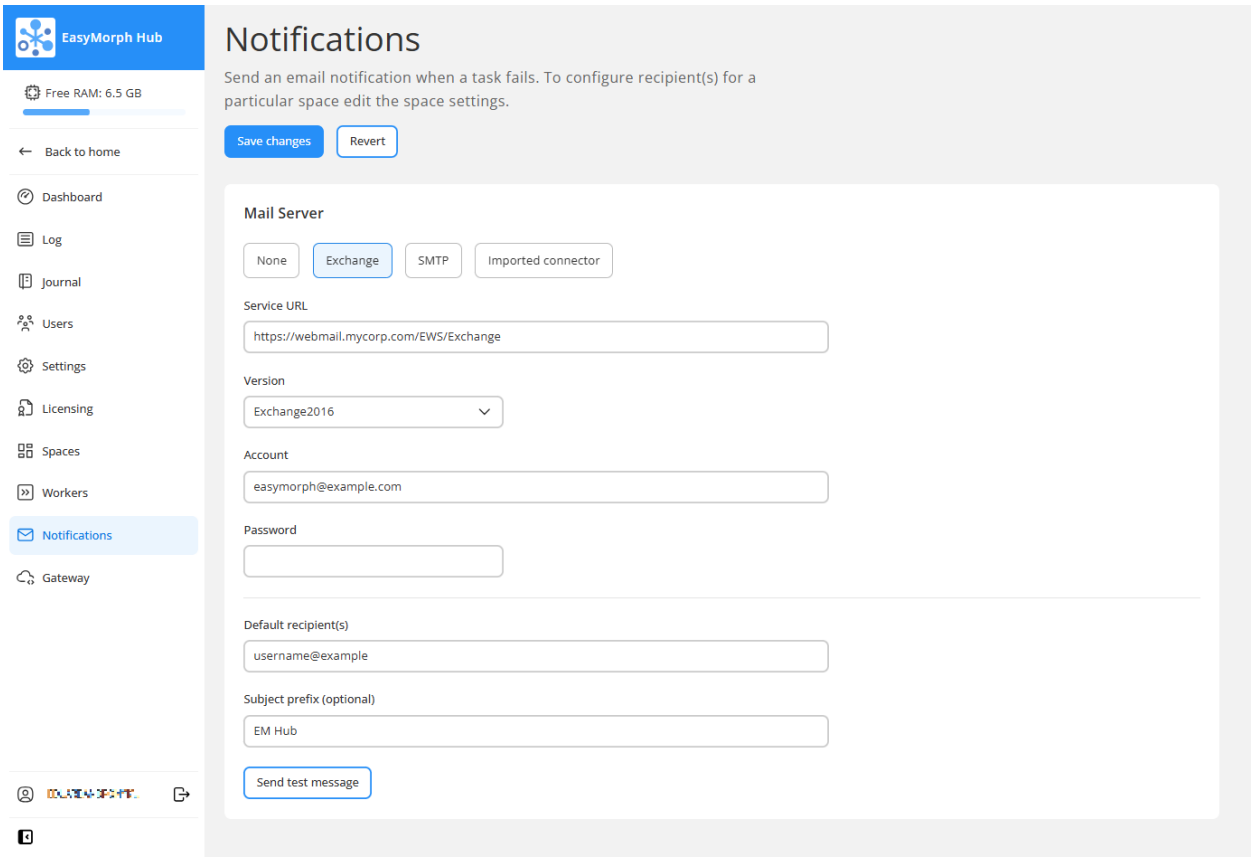
To re-connect a failed mapping (e.g. when the specified network location that was not available at the moment of Hub start), press “Edit” to open the mapping settings of the drive, and then press “Map” to initiate drive mapping. If mapping was successful the new drive becomes available right away.

To remove a mapped drive, press the trash bin icon of the mapped drive. The drive will be removed after the Server service is restarted (for the Default account), or when the worker is recycled (for other workers than Default).

Note that a mapped drive configured for a worker becomes available for all spaces that use the worker.

Email notifications about failed scheduled tasks

If the integration with an email service has been configured, EasyMorph Hub will automatically send notifications to a designated email address (or addresses) about failed scheduled tasks. If a task was triggered manually or through the API and failed, no notification will be sent.



Screenshot 26: Email notification settings.

Note that multiple default recipients can be specified – just separate their emails with commas.

Email notifications in spaces

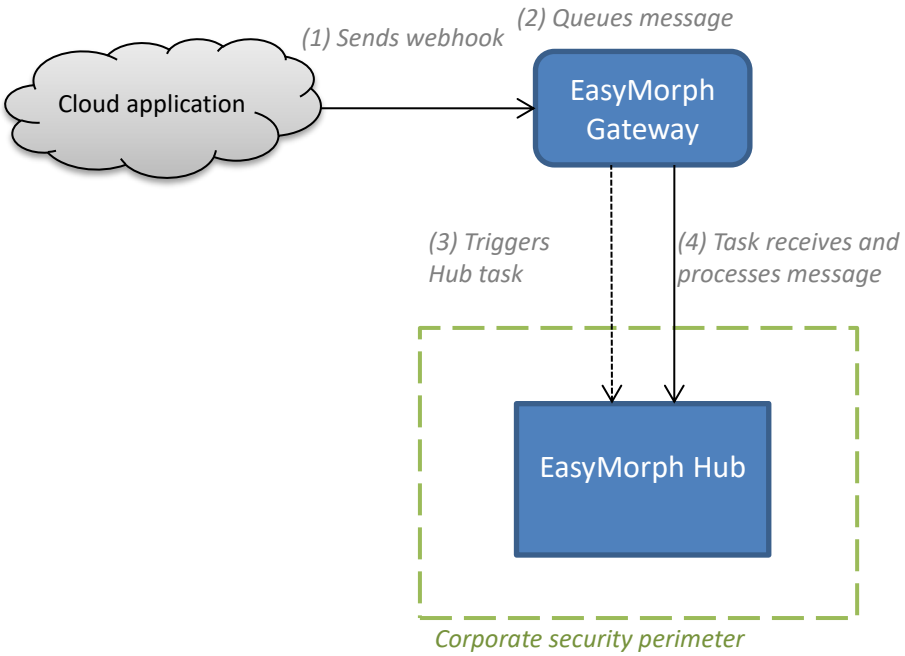
By default, all notifications about failed scheduled tasks are sent to the recipients stated in the Hub settings (tab Notifications). However, in space settings, it's possible to configure recipients (up to 10) that will receive notifications about failed tasks only in that particular space. In this case, notifications from this space won't be sent to the default recipient(s) configured in Hub settings.

EasyMorph Gateway

EasyMorph Hub can be linked to EasyMorph Gateway, an optional cloud service operated by the creators of EasyMorph and priced separately.

When Hub is linked to Gateway, it becomes possible to trigger Hub tasks from external and cloud applications via *webhooks* (many cloud applications offer them). Task parameters can be assigned with properties of incoming webhooks, such as request headers, URL parameters, request body, request method, etc.

You can also use Gateway endpoints to receive web-form submissions from a website and process them with Hub tasks.



Features of the Gateway:

- Highly available (built with a serverless technology)
- Industry-standard asymmetric end-to-end encryption
- Low latency (<2.5 seconds, 90th percentile)
- Up to 300KB of request body payload
- Optional authentication via API keys in request headers or URL parameters

Integration with Zapier and Power Automate

The Gateway also makes it possible to trigger Hub tasks from cloud automation services such as Zapier and Power Automate.

In Zapier, use the official EasyMorph action.

In Power Automate, add EasyMorph Hub as a custom target as described on the Community forum.

Connection to EasyMorph Gateway

The connection to EasyMorph Gateway is configured in the "Gateway" tab available to Hub administrators, as follows:

1. Generate a private key
2. Generate a public key for the private key
3. Send the public key and its hash to support@easymorph.com (technical support)
4. Configure the tenant ID and secret provided by the technical support team

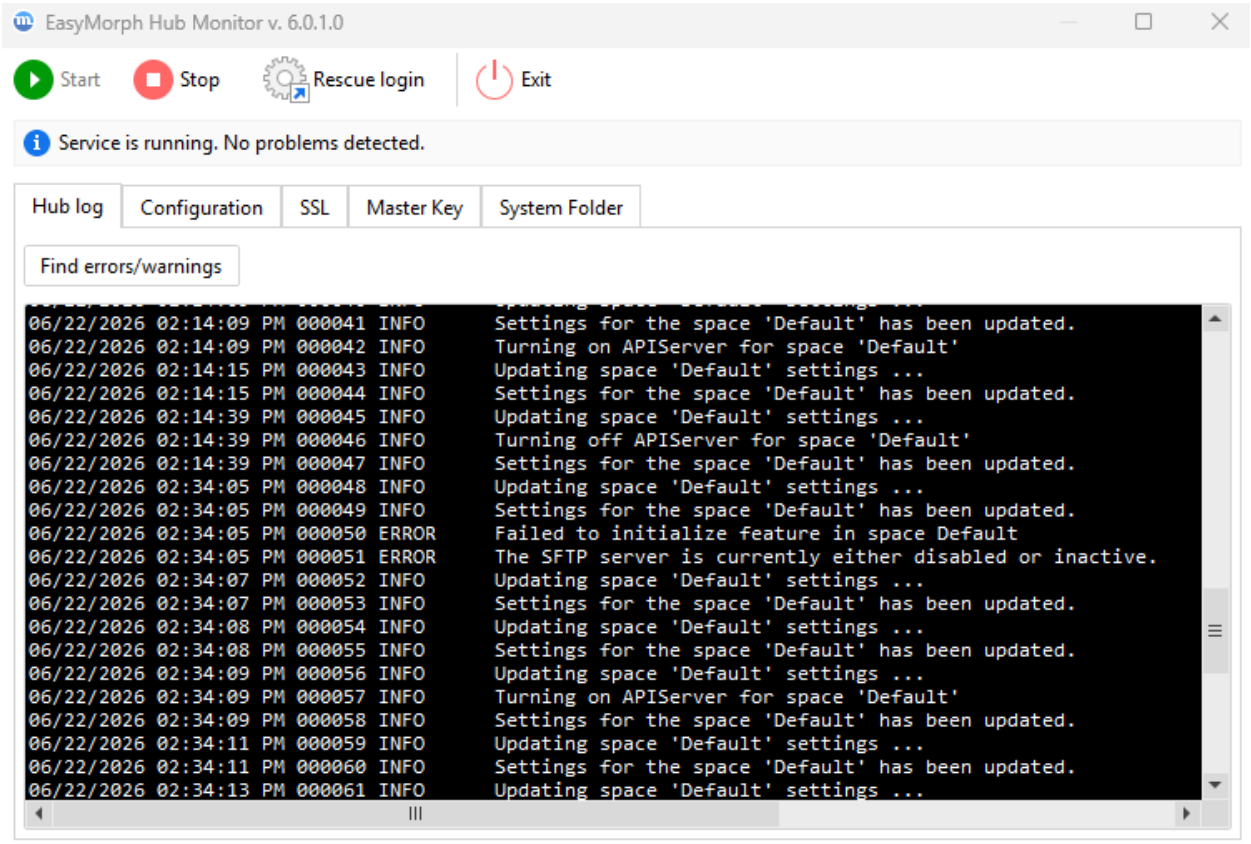
After the connection to Gateway is established, Hub administrators can create and configure public Gateway endpoints.

Important! Never share the private key from your Gateway configuration with anyone outside of your organization. Back up your private keys in a secure location. Decrypting your data coming through the Gateway is not possible without your private key.

Hub Monitor

EasyMorph Hub Monitor is a standalone Windows utility that is installed during Hub installation. It allows performing the following operations:

- Perform the password-less "rescue login" to access Hub settings
- See the service status (running/stopped/error)
- See the Hub log
- Start/stop the service
- Enforce/disable the SSL mode
- Change the service port (the service must be stopped prior to changing port)
- Import/export the master key of the Hub vault



Screenshot 27: EasyMorph Hub Monitor

The Monitor minimizes to the system tray. To exit the Monitor when it's minimized, right-click the tray icon and choose Exit.

Similarly to any other Windows service, EasyMorph Server service can also be started/stopped using the Windows Task Manager or Windows Services.

Backing up and restoring Hub

To back up and restore Hub configuration and data use a 3rd part disk backup utility. Make sure you export the Hub's master key before restoring from a backed-up disk image as explained below:

Hub vault

The Hub stores sensitive information such as authentication tokens and encryption keys in a secure vault. The vault itself is encrypted with a *master key* and locked to the current machine on the system level. Locking to machine means that the vault can't be decrypted on another machine, even if the whole disk image is restored from a backup. If the Hub can't access the vault, it's not operational. While locking the vault to machine provides an extra layer of security, it requires extra steps when performing backup/restore.

To enable backup/restore, export and import the master key as follows:

Backing up

1. Open the Hub Monitor utility (explained above) and export the master key into a password-protected binary file. This can be done once. No need to do it for every backup. Remember the password or store it in a secure store together with the master key file.
2. Create backup image(s) regularly.

Restoring

1. Have the password from the master key file ready.
2. Restore a disk image of the Hub.
3. Open the Hub Monitor utility (explained above) and import the master key file. Enter the password when prompted for.

HTTPS-only mode

The HTTPS-only mode uses the SSL encryption for all traffic between Hub and clients (web-browser, Desktop, API clients). When the HTTPS-only mode is turned on, all Hub URLs will start from **https://** instead of **http://**.

To switch the Hub to the HTTPS-only mode perform the following actions:

1) Open the Hub Monitor utility.
2) Stop the service.
3) Go to the SSL tab in the Monitor and tick “Use HTTPS”
4) Pick an SSL certificate from the list of installed certificates, or install a new certificate. Press Apply.
5) Start the service.

Note that:

- SSL certificates that are installed using the Monitor are installed into the Windows certificate store
- Accessing the Hub by unsecure HTTP (without SSL) is not possible after the SSL certificate is enforced in Hub Monitor
- If a self-signed SSL certificate is installed on EasyMorph Hub, make sure to add an exception in the Hub Link configuration of EasyMorph Desktop.
- Hub Link in EasyMorph Desktop switches to HTTPS automatically; no action is required.

Security considerations

EasyMorph Hub is technically a web application therefore all security considerations relevant to web applications apply.

If you host EasyMorph Hub on a cloud instance (e.g. Amazon EC2, Azure, or Google Cloud), make sure it's not exposed to the public internet. Use a VPN, a firewall, or another networking solution to allow access to the Hub only for the authorized users.

Other security recommendations:

- Do not use embedded connectors in workflows
- Do not use the anonymous access mode in spaces
- Do not use the “Legacy” access mode with shared passwords in spaces
- If using Hub user accounts with passwords, make sure they are sufficiently strong and have at least 20 characters (check out this xkcd about creating long passwords)
- Disable File Manager unless it’s necessary
    - If File Manager needs to be enabled then disable uploading files unless it’s necessary
- Prohibit arbitrary code execution unless it’s necessary
- Disable execution of unsigned projects unless it's necessary
- Do not use the Default worker for spaces, use other workers; make sure that the Windows account(s) used for the other workers can't access system folders, including the system folder of EasyMorph Hub (effectively, disable for them accessing anything that the "EasyMorph Server" group can access).
- SSL is configured and enforced, SSL certificate is valid and not expired
- Hub administrators don't access user content in spaces (i.e. don't have a role in ACL), unless it's necessary

Start/stop batch scripts

It is possible to execute a custom batch script when the Server service is starting or stopping. Modify **onstart.bat** or **onstop.bat** located in C:\Program Files\EasyMorph Server\systemscripts accordingly.

Logging

The Hub writes a log file located in the log folder specified on the Hub settings page.

If the log folder is not accessible to the "EasyMorph Server" Windows account group or to Server service account (e.g. because of lack of permissions) the Hub won’t start and will record an application error in the Windows Events log.

Note that EasyMorph Hub doesn't delete old Hub logs automatically. The log retention policy to left to Hub administrators who can, for instance, set up a Hub task to delete/archive old Hub log files.

Data persistence and locality

The Hub settings, task and space settings are stored in XML files and therefore can be backed up and restored using 3rd party tools, if necessary (except for v5.8). Database connectors, the embedded event journal, and catalog metadata are stored in SQLite database files which also can be backed up and restored using the same 3rd party tools.

No user data is stored in the Hub system folders. No data is sent to the cloud or to EasyMorph Inc. All data transformations are performed by EasyMorph's in-memory engine exclusively in memory without creating a disk footprint.

Command-line API client

EasyMorph Hub comes with a command-line utility *ems-cmd.exe* which is an open-source cross-platform command-line API client for the Hub. The utility allows triggering tasks, uploading/downloading files, checking Hub status and performing other Hub operations from the command line. It can be used in batch scripts, or called from external applications as a way of integration with EasyMorph Hub.

The source code, installer for standalone deployments, and documentation is available on GitHub: https://github.com/easymorph/server-cmd

The command-line API client is built using the .NET SDK (see below).

EasyMorph Hub .NET SDK

EasyMorph Hub can be programmatically integrated with 3rd party .NET applications using the .NET SDK available as a Nuget package. Its source code is open and available on GitHub too: https://github.com/easymorph/server-sdk

Migration to another machine

See this Community topic for the instruction on migrating EasyMorph Hub to another machine: https://community.easymorph.com/t/how-to-migrate-easymorph-server-to-another-machine/2194

Uninstallation

To uninstall EasyMorph Hub run the uninstaller and follow instructions. The uninstaller does NOT automatically remove:

- Task configuration files
- Hub configuration files
- Connector repositories
- Logs
- Embedded journal database(s)

They should be removed manually, if required.

Desktop to Hub Link

EasyMorph Desktop can be linked to EasyMorph Hub by configuring the Hub Link on the Home screen of EasyMorph Desktop. When the link is configured, it can be used by Desktop users for:

- Publishing projects to Hub
- Opening projects from Hub
- Publishing datasets to Hub
- Receiving datasets from Hub (including secure hot-linking)
- Using the repository of a Hub space in Desktops
- Leasing a user license from Hub
- Working with the Explorer

See more on configuring the Hub Link in this tutorial article: "Hub Link".

Notes:

- When a Desktop user uses a Hub-hosted connector, the connection to the source system (e.g., a database) is established from the user's computer, not from the Hub. No data is transmitted between Hub and Desktop when a Hub-hosted connector is being used on a Desktop.
- When EasyMorph Hub is switched to using SSL encryption (see chapter "HTTPS-only mode" above), Desktops will automatically switch from HTTP to using HTTPS in communications with the Hub. Therefore, no re-configuration of Hub Links in Desktops is necessary. To make EasyMorph Desktop always use SSL encryption (and disable using unsecured HTTP) when communicating with Hub, tick the "Require SSL" checkbox in the Hub Link configuration in Desktop.

UI customization

Starting from version 6.0, EasyMorph doesn't support custom themes. Instead, use the available customization settings of the Home page:

- Logo
- Background color of the logo
- Banner image

- Links below the banner image

Troubleshooting

Symptom	Action
Web Console inaccessible	Check if EasyMorph Hub service is running. If not then start it using EasyMorph Hub Monitor or Windows Services.
Service doesn't start	Use EasyMorph Hub Monitor to check the Hub log (default location is C:\Program Data\EasyMorph Server\Logs\Server log) for errors. If no errors in the Hub log, then check system events using Windows Event Viewer. Note that the Hub doesn't start when the log folder is not accessible for the Server service.
Hub doesn't recognize a license key	Make sure that the license key is located in a folder that has necessary permissions set up for the "EasyMorph Server" Windows account group. Folder C:\Users\Public might work.
Hub can't access a local folder	Make sure that the folder has the necessary permissions set up for the "EasyMorph Server" Windows account group or the worker that runs the task that can't access the folder.
Hub can't access a mapped drive	Re-connect the failed mapped drive in the worker settings. See chapter "Mapped drives".

Technical support

The EasyMorph Community forum is the main support channel and has many questions about Hub administration already answered.

If your request contains sensitive information that can't be published in a public forum, please contact our technical support at support@easymorph.com.