Parameters

It is possible to view and change the general server/application settings on "Parameters" page.

Two buttons are located in the upper part of the page:

- **Save** enables to save introduced changes of parameters.
- **Update** enables to upload currently used parameters.

> **Note:** It is required to press *Save* button upon completion of parameters correction.

Folders

Parameters of this group set directories for file storage in the server platform editions. The introduced changes take effect after the server service restart.

This group of parameters is not available in the desktop platform editions.

Parameter	Default value	Description
User data	UserStorage	The parameter is required for the name of the directory in which private directories of users created in Shared folders are located
Session backups	Session Backup	The parameter enables to set the name of the directory for session backups

Physically, the following path is used for location of all user folders on the server: `"C:\ProgramData\Megaladata\Server\UserStorage"`. This path can be changed in the configuration files of the Megaladata server. Apart from access to the personal folder, a user can be provided with access rights to other file storage folders by the server administrator.

The personal user folder and his account (login) have the same name.

The path to the file or package in the file storage is set relative to the UserStorage folder. For example:

- full path to the package: `"C:\ProgramData\Megaladata\Server\UserStorage\<user>\test.mgp"` where `<user>` is a user login;
- path to the package in the file storage: `<user>/test.mgp`.

LDAP Parameters

Integration with LDAP is available only in the server Enterprise edition.

Parameters of this group are not available in the server Team and Standard editions and in several desktop editions (Personal and Community).

Settings change for connection to LDAP server takes effect upon pressing 💾 *Save* button.

Parameter	Default value	Description
Host		Domain host name (or IP address) of the LDAP server. It is set by the empty string by default. In this case, the LDAP users are not checked.
Port	389	Number of the port for the LDAP server connection. 389 for the normal connection by default or 636 for the SSL connection.
Connection security	Normal connection	It enables to set secure (SSL/TLS) connection to LDAP via the special port (636 by default) or provide secure connection (STARTTLS) for the normal port via STARTTLS. Normal (nonsecure) connection by default.
Use SASL	false	Use SASL mechanism with DIGEST-MD5 password encryption for user authentification. It is used for the secure password propagation via nonsecure connection.
Account		User DN with LDAP read rights (search for users and groups). The account for LDAP connection must be located in the container specified in the Base Domain parameter. It is permissible to use domain\user format or UPN(user@domain.com) if the LDAP server makes it possible to do so. Empty name and password are used by default that provides the anonymous access to LDAP.
Password		Password for the read service account from LDAP
Base domain		Domain name of the directory `ou=users,dc=domain,dc=com` in which users are searched for

Parameter	Default value	Description
LDAP filter	(objectClass=user)	User filter in which it is also possible to verify user affiliation with a group. For example, filter for Active Directory can be `(&(objectClass=user)(memberOf:1.2.840.113556.1.4.1941:=cn=test,ou=groups,dc=domain,dc=com))` , for OpenLDAP `(objectClass=posixAccount)` where use of extension `:1.2.840.113556.1.4.1941:` for Active Directory enables to use mapping also by nested groups (it can be used only for Active Directory). Additional configuration can be required for filters in OpenLDAP.
Attribute with username	sAMAccountName	Attribute name in the user record that includes the username used for connection and automatic addition of Megaladata user when the LDAP authentication is successful. It is generally `sAMAccountName` for Active Directory, and `uid` for OpenLDAP.
Attribute with full username	displayName	Attribute name in the user record that includes the full username. The full username is used to specify the account owner in the Users list.
New user role	None	The parameter sets user role ("Workflows design" or "View reports") in the case of the automatic new user addition by means of LDAP authentication. By default, role is not assigned to the user created by means of LDAP. The user created with "Workflows design" role will be provided with rights to publish packages (refer to Users).

Security Parameters

Parameter	Default value	Description
Password entry attempt limit	Disabled	When user reaches this limit, he will not be able to log in anymore until *Password entry timeout* expires. History of failed password entry attempts is reset when Megaladata Server is restarted. If the maximum number of failed attempts of "Batch processing" account login is exceeded, batch processing using this account via Integrator and BatchLauncher will become unavailable before password entry timeout expiration. Minimum value - 1 attempt, maximum - 10000 attempts.
Password entry timeout (s)	∞	Period of restrictions imposed on the user who exceeded *Password entry attempt limit*. The restriction applies to all accounts, including the accounts with *Administration* and *Batch processing* roles. Minimum value - 1 second, maximum value - 2000000 seconds or ∞. If infinite timeout (∞) is set when password entry attempt limit is reached, the user will not be able to log in until Megaladata Server is restarted, or the administrator changes *Password entry timeout* or *Password entry attempt limit*.

Logging Parameters

Parameters of this group are set both in the server and desktop editions of the platform. They define log file generation rules. (It is a registration log in which execution of each node is fixed that enables to detect location and possible reason of the occurred error).

The introduced changes take effect upon pressing 💾 *Save* button.

Parameter	Default value	Description
Logging level	Information	Method and completeness of the log file filing is specified here (*Trace*, *Debug*, *Information*, *Event*, *Warning*, *Error* and *Fatal*)

Parameter	Default value	Description
Exception trace	false	It enables/disables debug operation mode. When the debug mode is enabled, the extended information is recorded to log file for error diagnostics. Generation of the extended information in the debug operation mode causes performance loss that is why exception trace must be disabled during the normal operation. The debug operation mode is disabled by default.
File name	app.log	The parameter contains the log file name. It is set by the path relative to Logs folder. By default, the file is located for the server edition in `"C:\ProgramData\Megaladata\Server\Logs"`, and for Desktop edition in `"C:\Users\User\AppData\Roaming\Megaladata\Personal\Logs"`.
Rewrite file	false	Checkbox of the logging file recreation while the application start instead of its supplementation. If the value is true, the log file will rewrite the old one at each start.
Maximum file size	10484760	The maximum size of the log file is set in bytes. When this size is exceeded, the file is saved with .1 extension, and the new one is created. For example, the old app.log file is saved as app.log.1 when this size is exceeded, and the new app.log file will be created instead of it where the log is maintained further.
Number of backup files	10	Number of backup copies created when the maximum size of the log file is exceeded. When the file size is exceeded, and the new logging file is created, the old file gets .1 extension with `+1: .2, .3 ... shift of the old saved files, .<maximum index>`, the old backup app.log.1 file will become app.log.2, and the oldest log file is deleted if the limit is reached at the moment of the backup file addition. When value < 1, old log files are not saved.

Connection Parameters

Connection parameter changes take effect after the server service restart.

Parameters of this group are not available in the desktop Megaladata editions.

Parameter	Default value	Description
Listen addresses	0.0.0.0	Address or addresses listened on the server availability
TCP port	4580	The port used for the server connection
WebSocket port	8080	The port used for message exchange with server
WebSocket SSL/TLS port	if WSS (WebSocket Secure) is configured, 8443, otherwise null	The secure port used for message exchange with server
Certificate file name		Path to the certificate file used for the SSL connection. It is set by the absolute or relative path (root directory `"C:\ProgramData\Megaladata\Server"`).

Parameter	Default value	Description
Private key file name		Path to the key file used for the SSL connection. It is set by the absolute or relative path (root directory `"C:\ProgramData\Megaladata\Server"`).
Connection check period (s)	Disabled	Time interval between check of connection with client. It is required for the automatic detection of connection loss and closure of corresponding client session with the packages opened in it. The state of the server and client connection channel is checked using the special message that is periodically sent to the client if no messages are sent to the server from the client during the set time interval. If any message is sent from the client, the timer is reset. Namely, the special message is sent only if no incoming message has been sent by the connection channel for a long period of time. It is possible to disable check of server and client connection. For this purpose, it is required to delete the check period value, save settings and restart the server. The disabled connection check can cause locking of packages by the server as the server will not be able to detect the loss of connection with the client, and it will be decided that the package is still being used by the client. In the case of the enabled parameter or small timeout, it can be impossible to reestablish connection after the connection loss.

Possible port values are set in the numeric format in the interval from 0 to 65535.

Server Parameters

Changes of the parameters that define the server operation mode, take effect upon pressing 💾 *Save* button.

> **Note:** The group of parameters is called *Application parameters* in the desktop editions, and it includes only *Size of thread pool* and *Locale* parameters.

Parameter	Default value	Description
Size of package pool	10	The number of the Megaladata Integrator packages frequently used in the batch processing saved for the quick call
Size of thread pool	By default	The maximum number of the threads inactive after use that are not deleted and kept in memory for possible later use. Default value — *Number of CPU cores* * 32. Maximum value — 1 000 000. Parameter increase secures faster operation against the background of many parallel operations. Parameter decrease means potential decrease of memory consumption.
Server stop timeout (s)	10	The time allocated to the server to complete operation. The minimum value is equal to 0.
Locale	Not set(...)[1]	It sets locale of the *Megaladata* server/application. Locale is used only after restart of the *Megaladata* server/application

[1]. Locale is set in the installation process in the server editions. It is set during the first start according to the regional settings of the current user in the desktop editions. ↩

Important: After locale change some nodes, for example, Sort, Export to the Text File, etc. can function in a different way.

Component Parameters: Program Execution

The introduced changes take effect upon pressing 💾 *Save* button.

Parameter	Default value	Description
Execution forbidden	true	The parameter forbids execution of the *Program execution* node if true value is set, and enables to execute the node when the value is false

> **Note:** *Program Execution* is allowed by default in the desktop editions. If required, it can be forbidden.

Component Parameters: Python

The introduced changes take effect upon pressing 💾 *Save* button.

Parameter	Default value	Description
Execution forbidden	true	The parameter forbids execution of the *Python* node if true value is set, and enables to execute the node when the value is false
Library path		It defines the path to python3x.dll file where x — number of minor release of the shared Python library. It is optionally specified. By default, the latest installed Python version is used. It is registered in the operating system. If path is not set in the parameter, it is automatically searched for in the environment variable `PYTHONHOME` and register. The easiest way to define path to python3x.dll library is to use command line in Windows. For this purpose, it is required to open "Command line" console ("CMD") and enter "where python*.dll" request (Python 3.9 version was selected as display example). The answer will be full path to the required library.

Parameter	Default value	Description
Interpreter path		It defines path to the executed interpreter file that executes script from the *Python* node in the Python execution mode in the separate process. Besides, this path is taken into account in the Python node execution during search for modules inside the Megaladata process. By default, *Interpreter path* is set as follows: 1. in *Windows* it is similar to search for shared library; 2. in *Linux* as follows: home Python directory is defined: if the actual path is specified in the system in `PYTHONHOME` environment variable, it must be used, otherwise the home directory is equal to `/usr`, b. the executed interpreter file is searched for by the following path relative to the home directory: `/bin/python3`, в. if the executed file is not found, python3 is executed without path specification.
Pass node environment variables	false	The parameter is required for the Python process start in the isolated environment. Environment variables are added to the Python process if the value is true: `LAUNCHER_PATHS` — the paths mounted to the file storage, shared `:` (Linux) or `;` (Windows), `LAUNCHER_USER` — Megaladata username, `LAUNCHER_USER_TMP` and `LAUNCHER_COMMON_TMP` — folder paths to save temporary files of user and work directory, `LAUNCHER_PIPEDIR` — name of the folder in which files of the named pipes for IPC (Linux) are created, `LAUNCHER_PIPENAME` — name of the duplex named pipes for IPC (Windows), `LAUNCHER_CURRENTDIR` — name of the Megaladata current folder (it matches the saved package path), `LAUNCHER_LOCALE` — name of the current locale (for example, `ru_RU`), `LAUNCHER_NODE_GUID` — node GUID, `LAUNCHER_NODE_NAME` — node name, `LAUNCHER_NODE_DISPLAYNAME` — node caption, `LAUNCHER_PREVIEW` is set for Preview.

For *Python* nodes operation, it is required to take into account the following peculiarities and restrictions:

- Minimum supported Python version — 3.5.
- Maximum tested Python version — 3.10.
- Creation of GUI interface from the Python code is not designation of the *Python* component, and in some cases it can cause incorrect application operation. For example:
  - node operation will not be terminated until GUI application operation is completed. As a rule, the Megaladata user does not have access to the GUI application window when working in the server editions.

For *Python* nodes operation in the isolated user environment, it is required to perform the following operations:

- It is required to enable "Pass node environment variables" option in the Administration section in the Python component parameters.
- It is required to specify full path to bash script in "Interpreter path" parameter — **python_run.sh** (or similar).
- *Docker* or *Podman* must be installed for start in Linux. In **python_run.sh** there is `DOCKER_OR_PODMAN` parameter responsible for containerization method, default value — `podman`.
- The base docker image with Python and required Python packages must be prepared. In **python_run.sh** image name is recorded in `IMAGE_NAME` parameter, default value — `python`.

In **python_run.sh** there is `CONTAINER_OPTIONS` parameter in which it is possible to pass container settings, for example, value of `oom-score-adj` variable. `oom-score-adj=1000` value is set by default. Namely, if it is required for OOM Killer to terminate a process, the container will be the first candidate for it.

For *Python* nodes operation inside the Megaladata process, it is required to take into account the following peculiarities and restrictions:

- It is required to install Python in the system with the same bitness as the Megaladata application/server bitness.
- Python from Anaconda distribution kit is not supported.
- multiprocessing package modules are not supported.
- Creation of GUI interface from the Python code is not designation of the *Python* component, and in some cases it can cause incorrect application operation. For example:
  - for `matplotlib` operation with Qt backend and in general with PyQt, availability of qt.conf file with paths to Qt libraries/plugins is required. If it is not possible to find them, the Megaladata application will be forcibly terminated with an error message.
- The background threads left unterminated can cause application crash.
- In the case of crytical errors python3x.dll can shut the application down. It is recommended to use preliminarily debugged Python code.
- Only one *Python* node can be simultaneously executed.
- It is possible to stop the *Python* node execution only between interpreter instructions.
- Virtual environments can be used when setting *Interpreter path*. For this purpose, the interpreter path must lead to python.exe inside the virtual environment more detailed information on search for Python modules inside the Megaladata process. To provide the correct operation, python.exe version in the virtual environment must match python3x.dll version that is used for the node code execution.

> **Note:** The *Python* node execution is allowed by default in the desktop editions. If required, it can be forbidden.
> **Note:** The Python execution mode is not supported in Linux inside the Megaladata process, consequently, there is no *Library path* setting. The *Python* nodes configured for execution inside the process will be executed in the separate process in Linux.

Search for Python modules inside the Megaladata process

Path to the folder in which interpreter is located is added as one more path in `sys.path` — list of the paths by which modules are searched for.

If for one directory above *Interpreter path* directory there is a file with `pyvenv.cfg` name, `sys.prefix` and `sys.exec_prefix` are set as equal to this directory, and availability of `site-packages` is checked. Found `site-packages` are added to `sys.path`. If `pyvenv.cfg` contains `include-system-site-packages` key and its value is not equal to true, the base path `site-packages` (relative to `sys.base_prefix`) is not included into `sys.path`.

Articles in Section:

- OpenID
- Logging
- Program Execution