



White Paper

Fabasoft Folio Environment Variables

Fabasoft Folio 2017 R1 Update Rollup 2

Copyright © Fabasoft R&D GmbH, Linz, Austria, 2018.

All rights reserved. All hardware and software names used are registered trade names and/or registered trademarks of the respective manufacturers.

No rights to our software or our professional services, or results of our professional services, or other protected rights can be based on the handing over and presentation of these documents.

Contents

1 Introduction	4
2 Software Requirements	4
3 General Notes	4
4 What Are Environment Variables?	4
5 Environment Variables in Fabasoft Folio	5
5.1 Environment Variables That Can Be Defined via the Microsoft Windows Registry or the Command Line	5
5.2 Environment Variables That can Only be Passed as Command Line Parameters	19
6 Environment Variables in Linux	20

1 Introduction

This document describes the usage of environment variables to specify default settings in the operating system. Settings like this can be defined in the command line, in the process environment or in the registry.

Environment variables provided in a Fabasoft Folio base installation are listed and described.

2 Software Requirements

System environment: All information contained in this document implicitly assumes a Microsoft Windows or a Linux environment.

Supported platforms: For detailed information on supported operating systems and software see the software product information on the Fabasoft distribution media.

Descriptions in this document are based on the following software:

Fabasoft Folio Backend Services

- Microsoft Windows Server 2016 Standard
- Red Hat Enterprise Linux 7.5 (x64)

3 General Notes

Variables can have different values, depending on the type of the variables. In case of the type "Boolean" the value can be `TRUE/FALSE`, `YES/NO` or `1/0`.

If registry entries are used to set variables, choose the type `REG_SZ` (string value) also for variables that contain numerical values. The declaration of numerical values can be made decimal or hexadecimal.

4 What Are Environment Variables?

Using environment variables basic settings in the operating system can be made. Such settings can be determined in various ways:

- in the command line
- in the process environment
- in the Microsoft Windows Registry (Microsoft Windows environment).

These levels are organized hierarchically, so that settings can be made in all levels. The declaration in the command line has the highest precedence. Then the value of the environment variable in the process environment is considered. If there are no declarations in these two levels, the entries in the Microsoft Windows Registry are used.

Often it is necessary to overwrite existing settings. This can be done via the command line, or, if this is not possible, via a `SET` command.

5 Environment Variables in Fabasoft Folio

As already mentioned, environment variables can be specified and determined via different methods. Many of the mentioned and described variables in this document can be used as command line parameters as well as Microsoft Windows Registry entries.

Environment variables that are specified in the Microsoft Windows Registry are located in the following key:

```
HKEY_CURRENT_USER\Software\Fabasoft\Environment
```

There are numerous variables that determine e.g. saving paths, connection settings and so on.

It is important to be aware of the correct notation when using variables as Microsoft Windows Registry entries or in the command line:

- Entries in the Microsoft Windows Registry are written in capital letters, e.g. `HOOSTCOOST`.
- If the variable is passed as parameter in the command line, the following syntax is valid:
`-host`
In this situation the usage of small letters for the name of the variable is important.

There are often small differences in the designation of the variable, depending on where it is used.

5.1 Environment Variables That Can Be Defined via the Microsoft Windows Registry or the Command Line

`CACHEDIR`

Path of the directory where the zero downtime cache is stored.

`COMMITDIR`

Path of the directory where modifications of local transactions are stored (e.g. transactions in the local mode of the client).

Example:

```
C:\ProgramData\Fabasoft\COMMITDIR
```

`DOCDIR`

Path of the directory where the cached contents are stored.

Example:

```
C:\ProgramData\Fabasoft\DOCDIR
```

`LOCALDATADIR`

Path of the directory where data of the Fabasoft Folio Web Services can be stored.

In a Microsoft Windows environment the default value for the Fabasoft Folio Web Services is `C:\ProgramData\Fabasoft`. This directory is used as root directory for `DOCDIR`, `CACHEDIR` and `COMMITDIR`, unless these variables are defined.

`GLOBALDATADIR`

The path of the directory in this variable determines where data that is required for all users of the Fabasoft Folio Web Client on this machine can be stored. This directory is used as root directory for `INSTALLDIR`, unless this variable is defined.

`INSTALLDIR`

Path of the directory where files of software components can be stored during the installation.

`ENABLECASCACHEDIR (TRUE/FALSE)`

Determines whether the Fabasoft Folio Kernel use a cache for CAS areas.

`CASCACHEDIR`

The path of the directory in this variable determines where the cache of the CAS areas is stored. The default value of this variable is `GLOBALDATADIR/CASCACHEDIR`.

`CASCACHEMAXMB`

This variable defines the maximum size of the CAS cache identified by `CASCACHEDIR`. The default value of this variable is 1024.

`CASCACHECLEANUPINTERVAL`

This variable defines the time interval in minutes of the recurring clearance process of the CAS cache considering the value of `CASCACHEMAXMB`. The default value of this variable is 120.

`MMCTEMPCLEANUPINTERVAL`

This variable defines the time interval in minutes of the recurring clearance process of the MMC area temporary directory. The default value of this variable is 120.

`MMCREADVERIFYHASH (TRUE/FALSE)`

This variable defines if the hash should be verified when loading the content. This option only takes effect when the Fabasoft Folio Kernel is configured for direct MMC area access. The default value of this variable is `TRUE`.

`MMCWRITETHREADCOUNT`

This variable defines the maximum number of threads writing parallel to a CAS area. This option only takes effect when the Fabasoft Folio Kernel is configured for direct MMC area access. The default value of this variable is 12.

`MMCREADDIR_<areaname>`

This variable defines a preferred read file share for a dedicated MMC area, if multiple file shares are configured for direct MMC area access.

DIRECTORYSEPERATENAME

This variable serves to save a string in the environment that is used to structure directories (e.g. `CACHEDIR`, `COMMITDIR`).

HOSTCOOST (Microsoft Windows Registry)

`-host` (command line)

Hostname or IP address of the machine where the primary COO store of the Fabasoft Folio Domain is located. This is determined during the installation process. It is the machine to which the Fabasoft Folio Web Client connects when logging in.

In a Fabasoft Folio COO Store (component object store) structured properties of objects are stored using a Fabasoft Folio COO Service. Stores in a domain are used to store different objects, whereas the primary Fabasoft Folio COO Store stores objects such as administration objects (domains, services, stores, user, groups and so on), desks and software product licenses, as well as component objects that are created in this domain.

PORTCOOST (Microsoft Windows Registry)

`-port` (command line)

In this environment variable the TCP/IP port number of the Fabasoft Folio COO Service that is used by the primary COO store for physical data storage is declared.

DOMAIN

Instead of a host and a port, the domain to which a user should connect can be defined using the environment variable `DOMAIN`. Every Fabasoft Folio Backend Service of a domain registers itself in the Microsoft Windows Active Directory. Using the domain ID, the network address of the primary COO service can be determined.

HOMEDOMAIN

Using this environment variable, it can be explicitly defined in which domain the user object should be searched for when logging in. Setting this value can be useful for inter-domain sharing.

CURRENTDOMAIN

This variable contains the encrypted value of the domain to which the Fabasoft Folio Kernel on this machine was connected last. A connection to this domain is established at the next start unless another setting is made (see the `CACHEDDOMAIN` variable).

CACHEDDOMAIN (TRUE/FALSE)

The value of this variable is `TRUE` by default, which means that the Fabasoft Folio Web Client connects to the domain which is stored in the `CURRENTDOMAIN` environment variable next time it is started.

If this value of `CACHEDDOMAIN` is set to `FALSE` the value in the variable `CURRENTDOMAIN` is not considered.

ISLOCAL (YES/NO)

With the value of this variable it can be determined if the Fabasoft Folio Kernel is in local mode at the moment.

ENABLELOCAL (YES/NO)

Determines whether the Fabasoft Folio Kernel can switch to local mode without explicit disconnection or not. `YES` means that all objects are always loaded completely to the cache. Therefore the Fabasoft Folio Kernel can work in local mode without disconnection.

CHECKINSTALLATION

The value of this variable determines whether software components should be updated automatically or not.

- `YES`
Default value of this variable. Fabasoft Folio Kernel and software components are installed or updated and the value of the `INSTALLDIR` variable is set.
- `COMPONENTS`
Only software components are updated automatically. This mode is used if the trace kernel was installed.
- `NO`
Neither the Fabasoft Folio Kernel nor the software components are updated. In this case all `.DLL` files have to be in one directory or found via the system environment variable `PATH`.
- `KERNEL`
Only the Fabasoft Folio Kernel is updated automatically.
- `SIMPLE`
Neither the Fabasoft Folio Kernel nor the software components are updated. The "dll" files can be located in the component directory (environment variable `INSTALLDIR`).
- `SEARCH`
Fabasoft Folio Kernel and software components are updated where they are found in the local system and if they are not found they are installed in the default directory.

STATFLAGS (Microsoft Windows Registry)

`-statflags` (command line)

In the Fabasoft Folio Kernel, statistics about all objects which are accessed are created. Via the environment variable `STATFLAGS`, the different statistics can be displayed. The value can be declared in decimal or hexadecimal forms.

- `-statflags = 0`
Disables the statistics functionality
- `-statflags = -1 OR 0xFFFFFFFF`
Enables the statistics functionality

Variables which are set via `STATFLAGS`:

- `STAT_OBJECT` Object-related statistics
0x00000001
- `STAT_ATTRIBUTEDEFINITION` Attribute-related statistics
0x00000002
- `STAT_ACTION` Action-related statistics
0x00000004
- `STAT_ERRORMESSAGE` Error message-related statistics
0x00000008
- `STAT_GENERAL` General statistics
0x00000010
- `STAT_TRANSACTION` Transaction-related statistics
0x00000020
- `STAT_REFRESH` Statistics of the refreshes
0x00000040
- `STAT_LOAD` Statistics of the loads
0x00000080
- `STAT_METHOD` Statistics of the method calls
0x00000100
- `STAT_KERNEL` Statistics of the kernel calls
0x00000200
- `STAT_TOP` Additional display of the top 20 entries of statistics values
0x10000000
- `STAT_DETAILED` Additional display of all entries of statistics values
0x20000000
- `STAT_NONE` Disables the statistics functionality
0x00000000
- `STAT_ALL` Enables the statistics functionality
0xFFFFFFFF

By default, all settings except `STAT_OBJECT` are active.

Example:

`STATFLAGS=264` or `STATFLAGS=0x00000108` displays statistics concerning method calls and error messages.

`ENABLEDTM` (TRUE/FALSE)

With this option, the usage of the Fabasoft Distributed Transaction Manager (DTM) for the execution of distributed transactions (in case of Oracle Database and PostgreSQL) can be disabled. If this parameter is set to `FALSE`, no 2-phase-commit protocol is used for distributed transactions. By default, the setting in the property *Enable 2-Phase Commit* in the current domain is used.

`ENABLEDTC` (TRUE/FALSE)

With this option, the usage of the Microsoft DTC for the execution of distributed transactions can be disabled. If this parameter is set to `FALSE`, no 2-phase-commit protocol is used for distributed

transactions. By default, the setting in the property *Enable 2-Phase Commit* in the current domain is used.

LOCALDTC (TRUE/FALSE)

The Microsoft DTC is installed on the Fabasoft Folio Backend Server to which the single Fabasoft Folio Web Clients connect. In doing so, the installation on the single client machines is not necessary.

SPECIFICDTC

Using this variable, a machine where a DTC is installed and that is used for the execution of distributed transactions can be specified.

SEARCHDTC (TRUE/FALSE)

If this variable is set to `TRUE`, there is an automatic search for a DTC when distributed transactions are executed.

RANDOMDTC (TRUE/FALSE)

If multiple DTCs are installed in a network choose a DTC to be used for distributed transactions at random by setting this variable.

CONNDOMAINDISTRIBUTEDTX (TRUE/FALSE)

If this variable is set to `FALSE`, no 2-phase-commit protocol is used for distributed transactions across connected domains.

CACHEMODIFY (TRUE/FALSE)

If this variable is set to `FALSE`, information in the client cache cannot be modified.

CACHEPERSIST (TRUE/FALSE)

If this value is set to `FALSE`, the client cache can be read and modified, but the modifications are not saved.

SIMPLELOGIN (TRUE/FALSE)

The identification of users when logging in to the Fabasoft product environment works with different methods. If `SIMPLELOGIN` is set to `TRUE`, only the Microsoft Windows account is considered.

This variable is also set to `TRUE` when calling a Fabasoft Folio Web Service. The method used to modify the values of variables is `SetKernelFlags()`.

EXTERNALUPGRADE (TRUE/FALSE)

This variable determines whether the file `cooupgr*.dll` is used during an upgrade of the Fabasoft Folio Kernel or not.

COMPLETE (TRUE/FALSE)

During the installation of a Fabasoft software product single software components can be chosen. This means that a complete installation does not have to be executed.

However, if the value of the variable `COMPLETE` is set to `TRUE`, all selected software products are installed completely.

SAFEMODE (TRUE/FALSE)

If the value of the variable `SAFEMODE` is set to `TRUE`, only methods of the software components of the domain 1.1 and 1.1001 are executed. The `SAFEMODE` variable can only be executed when using the trace kernel.

ERRORMODE

With this variable it can be specified which errors cause entries in a "log" file.

- `Normal`
Only grave errors are logged.
- `Off`
No errors are logged.
- `All`
All errors are logged.

ERRORLOG

Path of the file in which errors are logged.

TRACEMODE

This mode serves to log activities of the Fabasoft Folio Kernel or to debug own software components.

- `Normal`
Important trace messages are displayed in a message box.
- `Silent`
There is no display of trace messages in a message box.

TRACEFLAGS

The default trace flags used for all software components. The flags can be combined using a separator (e.g. space, comma, semicolon or pipe).

- `all`: Enable all traces.
- `calls`: Enable tracing of calls.
- `errors`: Enable tracing of errors.

- `expressions`: Enable tracing of expressions.
- `none`: Disable all traces.

`TRACEEXIT` (TRUE/FALSE)

The value of this variable determines whether trace messages should be displayed when closing the Fabasoft Folio Web Client or not.

`TRACELOCK` (TRUE/FALSE)

If this variable is set, trace messages are displayed by the thread that is responsible for the management of locks.

`TRACECACHE` (TRUE/FALSE)

This variable determines if the thread that is responsible for the management of the client cache displays trace messages.

`CACHESIZE`

This entry stores the value of objects that can be saved in the cache at maximum. The default value of the variable `CACHESIZE` for the Fabasoft Folio Kernel is 20000 objects. The default value for a Fabasoft Folio Web Service is 20000 objects for a 32-bit installation and 50000 objects for a 64-bit installation.

`CACHECLEANUPTHRESHOLDTIME`

If the Fabasoft Folio Kernel performs an aggressive cache cleanup, a warning can be created in the event log after a defined duration. This variable contains the time interval after which the event log entry is created. Long and recurring periods of aggressive cache cleanups indicate a cache size limit that could be too small for the workload.

Default value: 60 s

`CACHECLEANUPRECHECKTIME`

If the Fabasoft Folio Kernel performs an aggressive cache cleanup, a warning can be created in the event log after a defined duration. This variable contains the time interval used for generating event log entries for an ongoing aggressive cache cleanup after the threshold time has been exceeded.

Default value: 600 s

The following variables are used for extended error search e.g. in problems with locked objects:

- `LOCKTIMEOUTSLEEP` 5 ms
- `LOCKTIMEOUT1` 50000 ms
- `LOCKTIMEOUT2` 2000 ms
- `LOCKTIMEOUT3` 4000 ms
- `LOCKTIMEOUT4` 200 ms

- LOCKTIMEOUT5 10000 ms
- LOCKTIMEOUT6 1000 ms
- LOCKTIMEOUTLEVEL 5000 ms
- LOCKTIMEOUTRESOLUTION 32 ms
- LOCKTIMEOUTMODE 0

AUDITENTRYLIMIT

The value of this variable determines the maximum number of entries in an audit log object.

Default value: 2500 entries

AUDITTIMEOUT

When logging an action, the value of this variable determines the period of time after which a new entry concerning this action is generated in the audit log object.

Default value: 60 s.

MAXMETHODRECURSION

Specifies the maximum number of possible recursions which can occur until an error message is displayed.

Default value: 256

LASTLANGUAGE

In this variable, the object address of the language object that was used by the last user of the Fabasoft product environment in his user environment is saved.

LASTMACADDRESS

This variable stores the MAC address of the network card. This is required for the MLI mode.

The following variables concern the execution and validity of transactions:

- TRANSACTIONRETRYCOUNT
This variable determines how often a transaction is repeated after an error.
Default value: 2
- TRANSACTIONRETRYMIN
Minimum time interval after which a transaction can be repeated.
Default value: 500 ms
Note: A transaction is repeated after a randomly chosen time interval in the range of the values TRANSACTIONRETRYMIN and TRANSACTIONRETRYMAX.
- TRANSACTIONRETRYMAX
Maximum time interval after which a transaction can be repeated.
Default value: 2000 ms
Note: A transaction is repeated after a randomly chosen time interval in the range of the values TRANSACTIONRETRYMIN and TRANSACTIONRETRYMAX.

- `TRANSACTIONSKEWLEVEL`
In some cases, especially with web transactions, it is possible that time at which the transaction ends is earlier than the time of the start of the transaction. The value in the `TRANSACTIONSKEWLEVEL` variable determines the difference between start and end time of a transaction in the case that the end time is earlier than the start time so that the transaction is valid.
Default value: 600 s (10*60 s)
- `TRANSACTIONDISCARDTIME`
Transactions can run over a long period of time. The value of this variable specifies a maximum time interval in which a commit of the transaction may be regarded as valid.
Default value: 172 800 s (48*60*60 s)
- `TRANSACTIONRECHECKTIME`
If a transaction is executed, the security settings of the user in whose context the transaction is running are rechecked after a certain time interval.
Default value: 21 600 s (6*60*60 s)
- `TRANSACTIONAUTOLOCK` (TRUE/FALSE)
If set to `TRUE` and an object that has not been locked should be modified, an automatic lock is applied to the object.
Default value: `TRUE`
- `TRANSACTIONFORCELOCK` (TRUE/FALSE)
If set to `TRUE` and an object that has not been locked should be modified, an error is generated. `TRANSACTIONFORCELOCK=TRUE` overrides `TRANSACTIONAUTOLOCK=TRUE`.
Default value: `FALSE`

The following variables deal with communication between the Fabasoft Folio Kernel and the Fabasoft Folio COO Service:

- `COORETRYCOUNT`
This variable determines how often an operation is repeated after an error.
Default value: 1
- `COORETRYMIN`
Minimum time interval after which an operation can be repeated again.
Default value: 100 ms
- `COORETRYMAX`
Maximum time interval after which an operation has to be repeated again.
Default value: 500 ms
- `COORETRYTIMEOUT`
If an error occurs during the execution of an operation after the time interval specified in this variable, the operation is not repeated again.
Default value: 5000 ms
- `COOMAXRECHECKTIMEOUT`
If a local service is unavailable, it is rechecked after this timeout.
Default value: 5000 ms
- `COOMAXGLOBALRECHECKTIMEOUT`
If a global service is unavailable, it is rechecked after this timeout.
Default value: 60000 ms

The following variables deal with the communication between the Fabasoft Folio Kernel and the Fabasoft Folio MMC Service:

- `MMCRETRYCOUNT`
Default value: 1
- `MMCRETRYMIN`
Default value: 100 ms
- `MMCRETRYMAX`
Default value: 500 ms
- `MMCRETRYTIMEOUT`
Default value: 5000 ms
- `MMCMAXRECHECKTIMEOUT`
If a local service is unavailable, it is rechecked after this timeout.
Default value: 5000 ms
- `MMCMAXGLOBALRECHECKTIMEOUT`
If a global service is unavailable, it is rechecked after this timeout.
Default value: 60000 ms

These variables can be compared with the variables for the communication between the Fabasoft Folio Kernel and the Fabasoft Folio COO Service.

The following variables deal with event log messages in the communication between the Fabasoft Folio Kernel and the Fabasoft Folio Backend Services:

- `RPCTHRESHOLDTIME`
When executing RPCs an information message (Slow RPC call) can be created in the event log after a defined duration of the RPC. This variable contains the time interval after which the event log entry is created.
Default value: 10 s
- `COOTHRESHOLDTIME`
If an RPC is sent to a Fabasoft Folio COO Service a warning can be created in the event log after a defined duration of the RPC. This variable contains the time interval after which the event log entry is created.
Default value: 20 s
- `MMCTHRESHOLDTIME`
If an RPC is sent to a Fabasoft Folio MMC Service a warning can be created in the event log after a defined duration of the RPC. This variable contains the time interval after which the event log entry is created.
Default value: 20 s
- `RPCTHRESHOLD SIZE`
When executing RPCs an information message can be created in the event log when the RPC exceeds a specified size of data. This variable contains the size of data when the event log entry is created.
Default value: 500000 bytes
- `COOTHRESHOLD SIZE`
If an RPC is sent to a Fabasoft Folio COO Service a warning can be created in the event log when the RPC exceeds a specified size of data. This variable contains the size of data when the

event log entry is created.
Default value: 2000000 bytes

- `MMCTHRESHOLD SIZE`
If an RPC is sent to a Fabasoft Folio MMC Service a warning can be created in the event log when the RPC exceeds a specified size of data. This variable contains the size of data when the event log entry is created.
Default value: 5000000 bytes
- `REFRESHBLOCKLEVEL`
If a large number of objects is refreshed an entry in the event log is created. This variable contains the number of objects when an event log entry is created. The event log entries can help to detect performance problems.
Default value: 5000
- `LOADBLOCKLEVEL`
If a large number of attributes is loaded an entry in the event log is created. This variable contains the value of the number of objects when an event log entry is created.
Default value: 5000

`REFRESHBLOCKMAX`

If a large number of objects is refreshed this is performed block by block. This variable contains the number of objects that are refreshed in one RPC (in one block) at maximum.

Default value: 25000

`LOADBLOCKMAX`

If a large number of attributes is loaded this is performed block by block. This variable contains the number of objects that are loaded in one RPC (in one block) at maximum.

Default value: 25000

`LOCALOBJECTSLIMIT`

With the help of this variable it can be defined how many objects are deposited in the *Commonly Used* list (in objects of the class *Working environment*).

Default value: 1000

`LOCALOBJECTSLEVEL`

This variable defines the limit value which determines whether the *Commonly Used* list (in objects of the class *Working Environment*) is resorted when objects are reused or if the order is retained. Basically, objects are added at the end of the list. If the maximum number of objects in the object list is already reached, the first object is deleted from the list. If the list contains less than `LOCALOBJECTSLEVEL` objects, and an object which is already in the list is reused there is no resorting. Otherwise, if the list is already filled with more objects than defined in the variable `LOCALOBJECTSLEVEL`, and an object placed in the first `LOCALOBJECTSLEVEL` objects is reused, then this object is ranked backwards.

Default value: 500

To arrange searching efficiently, values in the following variables are considered:

- `QUERYRESTRLEVEL`
When executing a search, the objects which are found are passed to the client and a check of the ACL is executed for each single object (if the user has the right to search for this object). Depending on the rights that are allocated in the ACL, the objects are either shown or refused for the concerned user. If the number of checked objects exceeds the value in the variable `QUERYRESTRLEVEL`, an additional check of the search's efficiency is executed. (see `QUERYRESTRFACTOR`).
Default value: 5000
- `QUERYRESTRFACTOR`
After the number of objects defined in the variable `QUERYRESTRLEVEL` is checked, additionally the efficiency of the search is checked. This works with following formula:
displayed objects * `QUERYRESTRFACTOR` < *refused objects*
If the number of objects that a user is not allowed to search for is bigger than the number of objects that a user is allowed to search for multiplied by the value of the variable `QUERYRESTRFACTOR`, an error message is displayed.
- `QUERYSUBQUERYMAX`
When a full-text search or a search containing a sub-query is carried out (for example, a query containing a SELECT statement within another SELECT statement or sub-queries for the optimization of back-links or referenced properties), and the number of matches is greater than the specified value in this variable then the Fabasoft Folio Kernel generates the error message `QLPERR_INEFFICIENTSUBQUERY`. The results that are returned until this error is generated can however be subsequently used.
Default value: 50000
- `QUERYTHREADMAX`
This variable defines the maximum number of parallel COO Service queries triggered by a single Fabasoft Folio query.
Default value: 4

The following environment variables serve as strings transferred during an automatic software update:

```
COOPREP_BOOTSTRAP
COOPREP_COMMAND
COOPREP_INSTALL
COOPREP_PREPARE
COOPREP_TITLE
COOPREP_UNINSTALL
```

```
CONNTIME
```

Timeout value when a connection from the Fabasoft Folio Kernel to the Fabasoft Folio Backend Services is established. If this time interval (declared in seconds) is exceeded before the connection is established, an error message is displayed.

Default value: 5 s

RESPTIME

To establish a valid connection between the Fabasoft Folio Kernel and the Fabasoft Folio Backend Services, first the kernel has to send a request to the backend services and then the backend services have to send a response to the kernel.

The time that elapses before the confirmation from the backend services is received by the kernel is checked, and the value of the variable `RESPTIME` is considered as the maximum time interval. If this is exceeded an error message is displayed.

Default value: 15 s

STAT

Path of the file where using appropriate software statistics data can be logged.

TRACE

Path of the file where, when using the trace kernel, different information is logged.

JAVAHEAPSIZEINIT

When using Java implementations the defined initial and minimum Java heap size is crucial for functionality and performance. The default value is 64m (corresponds to `-Xms64m`).

JAVAHEAPSIZEMAX

When using Java implementations the defined maximum Java heap size is crucial for functionality and performance. The default value is 256m (corresponds to `-Xmx256m`).

JAVAPERMHEAPSIZEMIN

When using Java implementations the defined initial and minimum Java permanent generation heap size is crucial for functionality and performance. The default value is 32m (corresponds to `-XX:PermSize=32m`).

JAVAPERMHEAPSIZEMAX

When using Java implementations the defined maximum Java permanent generation heap size is crucial for functionality and performance. In case of `OutOfMemoryError` occurrences with the exception message "`PermGen space`", consider increasing the value. The default value is 128m (corresponds to `-XX:MaxPermSize=128m`).

COOJAVA_JVMOPTIONS

In this variable arbitrary JVM options can be passed to the Java virtual machine.

INDEXCLIENTTIMEOUT

The value of this variable determines the duration after a "slow index object" or "slow index content" message will be written in the indexing logs while building the Fabasoft Mindbreeze Enterprise full-text index from Fabasoft Folio. The default value is 1000 ms.

The following variables only apply to Fabasoft Folio Web Services:

- `FSCVEXT_ENABLEUNSECURESERVICEAUTHENTICATION`
Set this variable to true if portal authentication against a web service should be possible using a non-encrypted connection. Although unencrypted traffic is enabled and no certificates are needed, Fabasoft Folio accepts requests using portal authentication from trusted hosts only. That means the IP address of a host which uses this portal authentication has to be specified in the portal configuration in Fabasoft Folio.
- `FSCVEXT_EVTINFOPROCSECS`
If the total computing time exceeds the value defined in this variable an information message is recorded in the event log.
Default value: 30,000 ms
- `FSCVEXT_EVTWARNPROCSECS`
If the total computing time exceeds the values defined in this variable a warning is recorded in the event log.
Default value: 60,000 ms
- `FSCVEXT_EVTINFOBYTESREAD`
If a great number of bytes is read, reaching a certain number of bytes an information message is recorded in the event log.
Default value: 100,000 bytes
- `FSCVEXT_EVTWARNBYTESREAD`
If a great number of bytes is read, reaching a certain number of bytes a warning is recorded in the event log.
Default value: 500,000 bytes
- `FSCVEXT_EVTINFOBYTESENT`
If a great number of bytes is sent, reaching a certain number of bytes an information message is recorded in the event log.
Default value: 1,000,000 bytes
- `FSCVEXT_EVTWARNBYTESENT`
If a great number of bytes is sent, reaching a certain number of bytes a warning is recorded in the event log.
Default value: 5,000,000 bytes

5.2 Environment Variables That can Only be Passed as Command Line Parameters

When starting the Fabasoft Folio Kernel, different options can be specified using command line parameters.

```
-host [proto]
```

With the parameter `-host` the hostname or the IP address of the machine where the primary COO store of the Fabasoft Folio Domain is located can be declared. The Fabasoft Folio Web Client connects to this machine at login.

By default, the INET protocol is used for the connection. If the communication should take place using another network protocol, this can be determined using the parameter `-host [proto]`.

`-port [proto]`

A network protocol used for communication can also be declared for the specification of the port number of the Fabasoft Folio COO Service which is used by the primary Fabasoft Folio COO Store to store data physically.

`-update (TRUE/FALSE)`

Repeated registration of files of the software components so that controls are entered in the Microsoft Windows Registry. This parameter is also used when performing a software update.

6 Environment Variables in Linux

The process environment of Fabasoft Folio Services on Linux can be adjusted through the file system. Each Fabasoft Folio Service has got an instance directory.

`/var/opt/fabasoft/instances/<instance>`

Each instance directory contains a sub directory `env` which maps the process environment based on files:

`/var/opt/fabasoft/instances/<instance>/env`

Each file in the directory represents an environment variable. The file name accords to the key and the content to the value.

Example:

`/var/opt/fabasoft/instances/<instance>/env/HOST`

Host name or IP address of the machine that contains the primary COO store of the Fabasoft Folio Domain (to be declared during installation).

To set the value of the environment variable to „localhost“, perform following steps:

```
su - fscsrv
cd /var/opt/fabasoft/instances/<instance>/env
echo -n "localhost" > HOST
```

The environment variables which are declared in the Microsoft Windows Registry can be set on Linux in the file system.

`/etc/fabasoft/settings/users/fscsrv/Software/Fabasoft/Environment`

This path accords to:

`HKEY_CURRENT_USER\Software\Fabasoft\Environment`

Example:

To set the environment variable `CHECKINSTALLATION` to the Value `no`, following commands can be executed.

```
su - fscsrv
cd /etc/fabasoft/settings/users/fscsrv/Software/Fabasoft/Environment
mkdir CHECKINSTALLATION
echo -n "no" > CHECKINSTALLATION/registry.default
```