iQ.Suite KeyManager 5.0

Fine-Tuning Parameters

Dokumentversion 2.0

Inhalt

1 Motivation 4

2 Frontend Fine-Tuning Parameters 5

2.1 Location 5

2.1.1 On Windows 5

2.1.2 On Linux 5

2.2 Configurable Parameters 5

2.2.1 kms.configuration.dashboardReloadPeriod 5

2.2.2 kms.configuration.enableExport 5

2.2.3 kms.configuration.axisTimeoutSeconds 6

2.2.4 kms.configuration.enableConnectorBind 6

2.2.5 kms.configuration.enablePublicCertificatesView 6

2.2.6 kms.configuration.allowSameEmailIdForMultipleUsers 7

2.3 Usage Hints 7

3 Backend Fine-Tuning Parameters 8

3.1 Location 8

3.1.1 On Windows 8

3.1.2 On Linux 8

3.2 Important Notes 8

3.3 Configurable Parameters 9

3.3.1 kms.configuration.crl.connectionTimeout 9

3.3.2 kms.configuration.crl.readTimeout 9

3.3.3 kms.configuration.reconnect.attemptNumber 9

3.3.4 kms.configuration.replicateFineTuningInDb 9

3.3.5 kms.jobs.autoRenewJobSleepTime 10

3.3.6 kms.jobs.requestsJobSleepTime 10

3.3.7 kms.jobs.cpuLoadJobSleepTime 10

3.3.8 kms.jobs. messageQueueJobSleepTime 11

3.3.9 kms.jobs.mainQueueJobSleepTime 11

3.3.10 kms.jobs.monitorJobSleepTime 11
3.3.11 kms.jobs.forgottenPasswordJobSleepTime 12
3.3.12 kms.jobs.maxRequestsPerTurn 12
3.3.13 kms.configuration.readonly 12
3.3.14 kms.configuration.jobsDebugInfo 13
3.3.15 kms.configuration.DontIssueNewCertificates 13
3.3.16 kms.configuration.tryToRebuildPrivateKeyChain 13
3.3.17 kms.configuration.sessionTimeout 14
3.3.18 kms.configuration.db.maxbackups 14
3.3.19 kms.configuration.km_AuditLogs 14
3.3.20 kms.jobs.statusUpdaterJobSleepTime 14
3.3.21 kms.configuration.ntp.servers 15
3.3.22 kms.configuration.ntp.port 15
3.3.23 kms.configuration.ntp.timeout 15
3.3.24 kms.configuration.smime.sort.keyusage 16
3.3.25 kms.configuration.smime.sort.enddate 16
3.3.26 kms.configuration.useFineTuningSettingsFromLocalFile 17
3.3.27 kms.configuration.cert.usage.decrypt.renew 17
3.3.28 kms.configuration.cert.usage.encrypt.renew 17
3.3.29 kms.configuration.cert.usage.sign.renew 18
3.3.30 kms.configuration.cert.usage.verify.renew 18
3.3.31 kms.configuration.daysBeforeAuditlogDelete 18

4 Über GBS 20

1 Motivation

iQ.Suite KeyManager has a number of configurable options that are not visible on the User Interface but may only be set in configuration files. These options provide both minor and major behavior changes and may be used by advanced users to enhance the behavior of the product.

This document describes the fine-tuning parameters which may be changed by advanced users.

2 Frontend Fine-Tuning Parameters

These fine-tuning settings affect only the behavior of the KeyManager Administration Interface (shortly ‘Admin UI’).

2.1 Location

2.1.1 On Windows

The fine-tuning configuration file is usually located under:
*<KeyManager-InstallDir>\keymanager\etc\**kms.fe.finetuning.conf***

2.1.2 On Linux

The fine-tuning configuration file is usually located under:
*<KeyManager-InstallDir>\etc\**kms.fe.finetuning.conf***

2.2 Configurable Parameters

2.2.1 kms.configuration.dashboardReloadPeriod

- Type: Integer
- Default: 3
- Units: seconds

Specifies how often the *Home* page on the Admin UI will be refreshed. This includes CPU and memory usage, license information, tenant certificates count, etc.

2.2.2 kms.configuration.enableExport

- Type: Boolean
- Default: true

When set to **false,** it removes the EXPORT button on *Company Certificates* page. Neither business administrators, nor tenant users may then export company certificates on the Admin UI.

2.2.3 kms.configuration.axisTimeoutSeconds

- Type: Integer
- Default: 60
- Units: seconds

Defines the maximum timeout allowed between the frontend and backend web service calls (SOAP over HTTP/S). It may be helpful to increase this value, for example, if a large XML file (several megabytes of size) is to be imported into the KeyManager. The import process is synchronous and it may take considerable amount of time before all data gets imported at the backend side. If the backend fails to send a web service response back to the frontend in time, the frontend will timeout the operation and it will display an error message to the administrator. By increasing the timeout, the frontend will wait longer for the backend to reply.

2.2.4 kms.configuration.enableConnectorBind

- Type: Boolean
- Default: false
- Since 4.2.1

Setting this option to **true** will enable the BIND TO button option on the *Company Certificates* page. Using this functionality, one or more company certificates, imported from PKCS#12 files, may be assigned to a configured KeyManager connector. This means that when these certificates expire, the KeyManager will automatically try to renew them using the specified connector.

> **Caution**: This option does not work with the VRIdent connector, because it requires additional meta data that KeyManager saves only during registration of VRIdent certificate requests. PKCS#12 files do not carry any such information, therefore binding is not possible.

2.2.5 kms.configuration.enablePublicCertificatesView

- Type: Boolean
- Default: true
- Since 4.5.0

Setting this option to **true** will enable the SMIME public certificate view page to the users. Using this option will allow user to search and find the list of available SMIME public certificates in the KeyManager. When this option is sets to **false**, then SMIME public certificate view page will be disabled for all users.

2.2.6 kms.configuration.allowSameEmailIdForMultipleUsers

- Type: Boolean
- Default: false
- Since 4.5.0

By default, when a new user is created in KeyManager, a unique email address must be provided for the user. When this parameter is set to **true**, the same email address can be used for multiple users. This makes setting up a single mailbox for multiple users easier.

> **Caution:** By setting this option to **true**, users might receive KeyManager messages that are not intended for them.

2.3 Usage Hints

The KeyManager needs to be restarted after editing the *kms.fe.finetuning.conf* fine-tunings file!

3 Backend Fine-Tuning Parameters

3.1 Location

3.1.1 On Windows

The fine-tuning configuration file is usually located under:
*<KeyManager-InstallDir>\keymanager\etc\**kms.be.finetuning.conf***

3.1.2 On Linux

The fine-tuning configuration file is usually located under:
*<KeyManager-InstallDir>\etc\**kms.be.finetuning.conf***

3.2 Important Notes

Please read the following information concerning the usage of backend fine-tuning settings.

- Backend fine-tuning settings are always read from the database, unless kms.configuration.useFineTuningSettingsFromLocalFile is set to **true**.
- When kms.configuration.useFineTuningSettingsFromLocalFile is set to **true**, the KeyManager uses only the fine-tuning settings defined in the local *kms.be.finetuning.conf* file. The fine-tuning settings saved in the database will not be used.
- To overwrite fine-tuning settings in the database with fine-tuning settings from the local *kms.be.finetuning.conf* file, set kms.configuration.replicateFineTuningInDb to **true**. This can be done at runtime without the need to restart the KeyManager.
- For every change in the *kms.be.finetuning.conf* file, unless it is the kms.configuration.replicateFineTuningInDb parameter, it is recommended to restart the KeyManager, so for the changes to take effect.
- When starting the KeyManager after an upgrade, newly available backend fine-tuning settings will automatically be added to the *kms.be.finetuning.conf* file.

3.3 Configurable Parameters

3.3.1 kms.configuration.crl.connectionTimeout

- Type: Integer
- Default: 10
- Units: seconds
- Since: 5.0.7

Set the maximum amount of time to establish the connection to CRL host. The connection will be aborted after this timeout expires.

3.3.2 kms.configuration.crl.readTimeout

- Type: Integer
- Default: 180
- Units: seconds
- Since: 5.0.7

Set the maximum amount of time to wait for a CRL to be downloaded. The download will be aborted after this timeout expires.

3.3.3 kms.configuration.reconnect.attemptNumber

- Type: Integer
- Default: 3
- Since: 5.0.7

Specifies how often the system tries to re-establish a lost database connection. After the number of trials have been exceeded, the system will shut down.

3.3.4 kms.configuration.replicateFineTuningInDb

- Type: Boolean
- Default: false
- Since: 2.0.0

Set this option to **true** to trigger replication of fine-tuning settings from the *kms.be.finetuning.conf* file to the database. This configuration option is useful if you have multiple KeyManager servers using the same database and you want the fine-tuning settings in the KeyManager *kms.be.finetuning.conf* file to be replicated to all other KeyManager server instances. Note that a KeyManager server will read and use the fine-tuning settings from the database, but it will not overwrite its local *kms.be.finetuning.conf* file.

For further information, please refer to kms.configuration.useFineTuningSettingsFromLocalFile.

> **Caution**: The replication process is **immediate**! This means that the KeyManager actively monitors the *kms.be.finetuning.conf* file and if the replication option is set to **true**, it will immediately replicate the fine tuning settings to the database and then set the *kms.configuration.replicateFineTuningInDb* option back to **false**. It is recommended to restart all KeyManager servers for the changes to take effect.

3.3.5 kms.jobs.autoRenewJobSleepTime

- Type: Integer
- Default: 600
- Units: seconds
- Since: 1.1.0

Sets in what time interval the certificate renewal job will run. This job will evaluate which certificates, in all existing tenants, need to be scheduled for renewal.

3.3.6 kms.jobs.requestsJobSleepTime

- Type: Integer
- Default: 30
- Units: seconds
- Since: 1.1.0

Sets in what time interval the certificate requests processing job will run. This job processes all types of operations related to S/MIME certificates, i.e. new certificate requests, certificate revocation, certificate renewals.

3.3.7 kms.jobs.cpuLoadJobSleepTime

- Type: Integer
- Default: 3
- Units: seconds

- Since: 1.1.0

Sets in what time interval the CPU usage job will run. This job determines the current CPU usage which the KeyManager shows on the *Home* page.

3.3.8 kms.jobs. messageQueueJobSleepTime

- Type: Integer
- Default: 20
- Units: seconds
- Since: 1.1.0

Sets in what time interval the message queue job will run. This job is responsible for sending KeyManager emails to administrators and users.

3.3.9 kms.jobs.mainQueueJobSleepTime

- Type: Integer
- Default: 10
- Units: seconds
- Since: 1.1.0

Sets in what time interval the main queue job will run. This job is responsible executing common KeyManager server tasks like OCSP checks and certificates chain rebuild.

3.3.10 kms.jobs.monitorJobSleepTime

- Type: Integer
- Default: 20
- Units: seconds
- Since: 1.1.0

Sets in what time interval the monitor job will run. This job monitors the backend state and the SQL server connections. If the SQL server is down, this job will bring the backend to a temporary error state in order to prevent data loss. When the SQL server is up again, the monitor job will reset the backend state and make it operational again. Users that were logged in prior to when the SQL connection was lost, will be disconnected and will have to log in again.

3.3.11 kms.jobs.forgottenPasswordJobSleepTime

- Type: Integer
- Default: 30
- Units: seconds
- Since: 1.1.0

Sets in what time interval the forgotten password job will run. This job monitors password reset requests and deletes all requests that have expired. Each password reset request is valid for 1 hour.

3.3.12 kms.jobs.maxRequestsPerTurn

- Type: Integer
- Default: 30
- Since: 1.1.0

Sets the amount of certificate requests, both new certificate requests and renewal certificate requests, the KeyManager may process in one pass. The rest of the certificate requests are placed in a queue and will be processed the next time the job runs. The job timeout is configured with the *kms.jobs.requestsJobSleepTime* configuration option.

> **Caution**: It is not recommended to set high values for this configuration option because certificate generation is a CPU consuming task!

3.3.13 kms.configuration.readonly

- Type: Boolean
- Default: false
- Since: 1.1.0

This is a debugging configuration option.

When set to **true**, the KeyManager will only save configuration changes to memory and not to the file system or to the database. When the backend is restarted, the configurations will be reset.

3.3.14 kms.configuration.jobsDebugInfo

- Type: Boolean
- Default: false
- Since: 1.1.0

This is a debugging configuration option.

When set to **true**, the KeyManager jobs will print more verbose logging messages in logs.

3.3.15 kms.configuration.DontIssueNewCertificates

- Type: Boolean
- Default: false
- Since: 1.1.0

When set to **true**, the KeyManager instance will not process certificate requests even if it finds requests that have been already approved. This configuration option should be used if there are 2 or more KeyManager server instances using the same SQL database server. In order to prevent conflicts, there must only be one KeyManager instance that is responsible for processing certificate requests. Certificate requests may be created on all KeyManager server instances, but may only be processed by one designated instance.

Caution: Use this option only when *kms.configuration.useFineTuningSettingsFromLocalFile* is also set to **true**! This is required, because otherwise the *kms.configuration.DontIssueNewCertificates* option may get replicated to the database and all KeyManager instances will stop processing certificate requests after restart.

3.3.16 kms.configuration.tryToRebuildPrivateKeyChain

- Type: Boolean
- Default: false
- Since: 3.0.0

When set to **true**, every time iQ.Suite requests a private key, the KeyManager will return the private key with the corresponding certificate chain, but only if the chain is already available in the KeyManager database. If set to **false**, or the certificate chain is not available, KeyManager will return only the private key. This feature is useful for iQ.Suite email signatures where, if configured, the iQ.Suite may attach the complete certificate chain to the signed S/MIME message.

3.3.17 kms.configuration.sessionTimeout

- Type: Integer
- Default: 600
- Units: seconds
- Since: 2.2.0

Sets the amount of time KeyManager users may stay idle while logged into the Admin UI. If there are no user actions done on the Admin UI for the configured period of time, the user will be forcefully logged out of the KeyManager.

3.3.18 kms.configuration.db.maxbackups

- Type: Integer
- Default: 10
- Since: 3.1.0

When set to **true**, KeyManager will delete the oldest H2 database backup file at boot time.

At boot time, KeyManager creates a backup of the last used H2 database in the configured `<KeyManager-InstallDir>\data` folder. This option defines how many backup files the KeyManager may create before starting to delete old ones.

3.3.19 kms.configuration.km_AuditLogs

- Type: Boolean
- Default: true
- Since: 3.2.0

When setting to **true**, KeyManager will log all user actions related to certificate management, triggered by users in the Admin UI. The audit log entries will be shown in the certificate audit UI and in the certificates log tab view. This log entries will be stored in the back-end database table for future reference.

When setting to **false**, KeyManager will not capture the certificate actions performed by the user and not make entries in the backend table.

Disabling this option may increase performance of the KeyManager when under heavy load.

3.3.20 kms.jobs.statusUpdaterJobSleepTime

- Type: Integer

- Default: 60
- Units: seconds
- Since: 3.1.0

Sets in what time interval the certificate status update job will run. This job evaluates which certificates, in the existing tenants, have expired and sets their status to *Expired*.

3.3.21 kms.configuration.ntp.servers

- Type: CSV List
- Default: de.pool.ntp.org,pool.ntp.org,ntp.ubuntu.com
- Since: 3.5.0

At boot time, KeyManager will attempt to connect to a Time server using the Network Time Protocol (NTP) and verify that the system time of the server where KeyManager is installed is up-to-date. If the list is empty, this step will be skipped.

3.3.22 kms.configuration.ntp.port

- Type: Integer
- Default: 123
- Since: 3.5.0

Sets the port number to be used for Time server NTP connections. NTP is an UD- based protocol.

> **Caution**: Port 123 is usually forbidden when behind a DMZ, therefore extra measures need to be taken by IT when this option is configured to work with external NTP servers.

3.3.23 kms.configuration.ntp.timeout

- Type: Integer
- Default: 30
- Units: seconds
- Since: 3.5

Set the maximum amount of time to wait for an NTP server reply. The UDP connection will be aborted after this timeout expires.

3.3.24 kms.configuration.smime.sort.keyusage

- Type: Boolean
- Default: true
- Since: 4.1.0

When iQ.Suite requests a certificate for a user and certain purpose (e.g. encryption), KeyManager uses a certificate selection algorithm to return the certificate that fits best to the request. If there is no certificate for the user that has an appropriate key usage, KeyManager returns no certificate by default.

However, if this parameter is set to **false**, KeyManager returns the certificate that fits best to the request, even if the key usage of the best certificate is not appropriate for the request.

> How the certificate selection algorithm in KeyManager behaves regarding the key usage can also be configured in the configuration of newer iQ.Suites. The configuration setting in iQ.Suite has precedence to this parameter.

3.3.25 kms.configuration.smime.sort.enddate

- Type: Boolean
- Default: true
- Since: 3.5.4

When iQ.Suite requests a certificate for a user and certain purpose, KeyManager uses a certificate selection algorithm to return the certificate that fits best to the request. By default, a certificate with a later validity end date is considered better than a certificate with a later validity start date. By setting this parameter to **false**, the behavior of the certificate selection algorithm is changed. Then certificates with a later validity start date will be considered better than certificates with a later validity end date.

> How the certificate selection algorithm in KeyManager behaves regarding the validity dates can also be configured in the configuration of newer iQ.Suites. The configuration setting in iQ.Suite has precedence to this parameter.

3.3.26 kms.configuration.useFineTuningSettingsFromLocalFile

- Type: Boolean
- Default: false
- Since: 3.5.3

Set this option to **true** to force the KeyManager to always read the fine-tuning settings from the local *kms.be.finetuning.conf* file and ignore the fine-tuning settings saved in the database. This configuration option is useful if you have 2 or more KeyManager servers using the same database and you want one of the KeyManager instances to have fine-tuning settings independent from the other instance. This option would normally be used in conjunction with kms.configuration.DontIssueNewCertificates set to **true**.

3.3.27 kms.configuration.cert.usage.decrypt.renew

- Type: Boolean
- Default: true
- Since: 4.5.0

The automatic approval of certificate renewal requests can be configured to depend on the last usage date of the certificate. This way certificates that are no longer used (e.g. because the corresponding user left the company) are not automatically renewed.

By default, a request from iQ.Suite for a certificate for decryption is considered as a usage of the certificate. Therefore, the last usage date of the certificate will be updated when the certificate is passed to iQ.Suite for decryption. If this parameter is set to **false**, decryption will not be considered as a usage of the certificate and the last usage date will not be updated.

3.3.28 kms.configuration.cert.usage.encrypt.renew

- Type: Boolean
- Default: true
- Since: 4.5.0

The automatic approval of certificate renewal requests can be configured to depend on the last usage date of the certificate. This way certificates that are no longer used (e.g. because the corresponding user left the company) are not automatically renewed.

By default, a request from iQ.Suite for a certificate for encryption is considered as a usage of the certificate. Therefore, the last usage date of the certificate will be updated when the certificate is passed to iQ.Suite for encryption. If this parameter is set to **false**, encryption will not be considered as a usage of the certificate and the last usage date will not be updated.

3.3.29 kms.configuration.cert.usage.sign.renew

- Type: Boolean
- Default: true
- Since: 4.5.0

The automatic approval of certificate renewal requests can be configured to depend on the last usage date of the certificate. This way certificates that are no longer used (e.g. because the corresponding user left the company) are not automatically renewed.

By default, a request from iQ.Suite for a certificate for signing is considered as a usage of the certificate. Therefore, the last usage date of the certificate will be updated when the certificate is passed to iQ.Suite for signing. If this parameter is set to **false**, signing will not be considered as a usage of the certificate and the last usage date will not be updated.

3.3.30 kms.configuration.cert.usage.verify.renew

- Type: Boolean
- Default: true
- Since: 4.5.0

The automatic approval of certificate renewal requests can be configured to depend on the last usage date of the certificate. This way certificates that are no longer used (e.g. because the corresponding user left the company) are not automatically renewed.

By default, a request from iQ.Suite for a certificate for verification is considered as a usage of the certificate. Therefore, the last usage date of the certificate will be updated when the certificate is passed to iQ.Suite for verification. If this parameter is set to **false**, verification will not be considered as a usage of the certificate and the last usage date will not be updated.

3.3.31 kms.configuration.daysBeforeAuditlogDelete

- Type: Integer
- Default: 1096
- Units: days
- Since: 4.5.0

With this parameter, the time of automatic deletion of audit log entries can be configured. By default, audit log entries are deleted after 1 096 days (ca. 3 years). A minimum value of 14 days and maximum value of 32 000 days is allowed. The backend audit log deletion job will run during KeyManager start up and every 24 hours when KeyManager is running.

If the value is set to less than 14 days, KeyManager will reset the value to the minimum of 14 days. To turn of the audit log, refer to kms.configuration.km_AuditLogs.

4 Über GBS

GROUP Business Software is a leading vendor of solutions and services in the fields of messaging security and workflow for the IBM and Microsoft collaboration platforms. Over 5,000 customers and more than 4 million users worldwide trust in GBS expertise. The company operates in Europe, North America and Asia.

© 2016 GROUP Business Software GmbH

Our product descriptions are of a general and descriptive nature only. They do not stipulate any specific features nor do they represent any form of warranty or guarantee. We reserve the right to change the specifications and design of our products without notice at any time, in particular in order to keep abreast of technical developments.

The information contained in this document presents the topics from the viewpoint of GROUP Business Software GmbH (hereafter 'GBS') at the time of publishing. Since GBS needs to be able to react to changing market requirements, this is not an obligation for GBS and GBS cannot guarantee that the information presented in it is accurate after the publication date.

This document is intended for information purposes only. GBS does not extend warranty for this document, in either explicit or implied form. This also applies to quality, execution, standard commercial practice or suitability for a particular purpose.

All the product and company names that appear in this document may be trademarks of their respective owners.