iQ.Suite Update Manager 7.0

Central Update Service for iQ.Suite Components in Multi-Server Environments

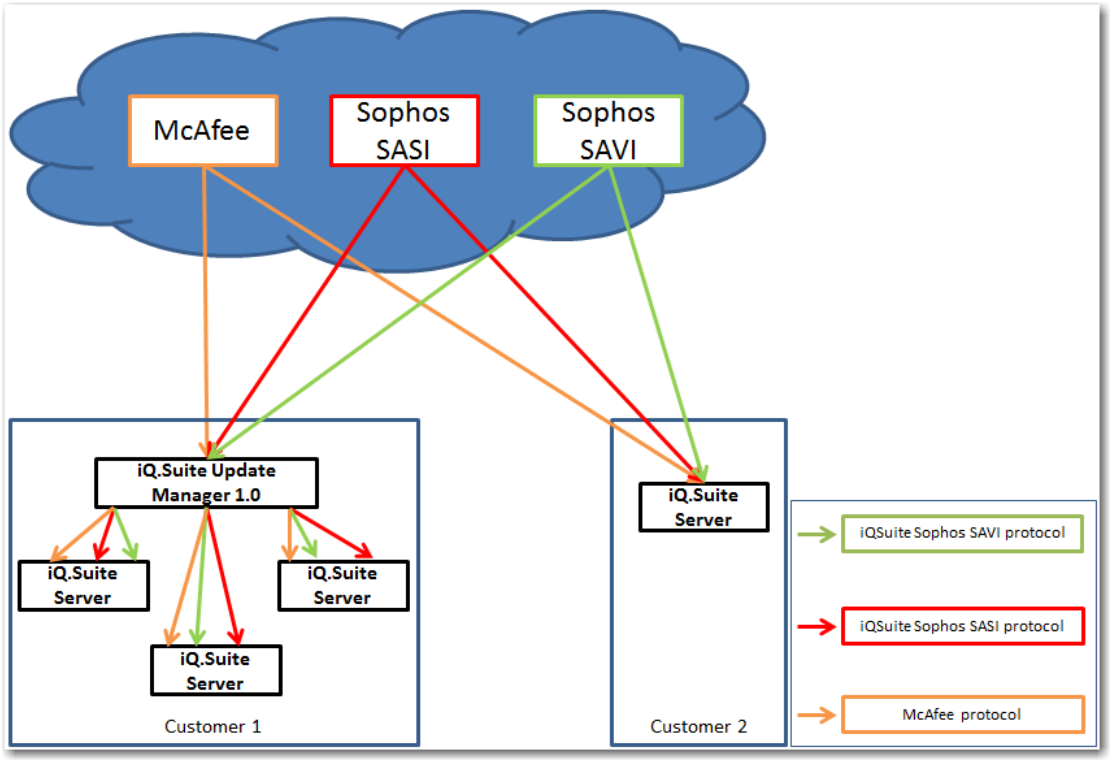
Document Version 12.0

iQ.Suite Domino
iQ.Suite Microsoft

Contents

1 Functionalities of iQ.Suite Update Manager

The iQ.Suite Update Manager can be used to automatically download the current Engines and patterns for virus scanning (SAVAPI, SAVAPI2, KAV, McAfee, SAVI) or for spam analysis (SASI, KAS) and to provide them via a central web server for the *iQ.Suite for IBM Domino* and *iQ.Suite for Microsoft Exchange/SMTP*. This way, all used iQ.Suite installations can get their updates from this web server through your enterprise internal network. This results in saving of bandwidth and costs (Customer 1 in the illustration). In addition, iQ.Suite Update Manager provides the Engines and patterns centrally also in mixed Domino and Exchange environments.



Note: Only some of the OEM products supported are represented as examples in the illustration above.

Supported OEM products[1]**:**

- Avira SAVAPI (**S**ecure **A**nti**V**irus **A**pplication **P**rogramming **I**nterface)
- Avira SAVAPI2, 32 bit and 64 bit
- KAV (**K**aspersky **A**nti-**V**irus), 32 bit and 64 bit
- KAS (**K**aspersky **A**nti-**S**pam)
- McAfee Virus Scanner, 32 bit and 64 bit

[1] These products can be selected as iQ.Suite components during setup.

- SAVI (**S**ophos **A**nti-**V**irus **I**nterface)
- SASI (**S**ophos **A**nti-**S**pam **I**nterface)

Only Kaspersky & SAVAPI / SAVAPI2:
Providing License Files for the Scanner Usage

In addition to the actual scanner updates, the required license files for Kaspersky Anti-Virus / Anti-Spam and SAVAPI / SAVAPI2 can be downloaded and provided. Moreover, Update Manager can thereby update its own used license files.

Update Validation before Providing Files (CheckFiles)

Validation of the downloaded files can be enabled for all supported products separately. With this, the downloaded files are first checked for validity via an executable file (BAT, EXE, CMD, or similar). Only the valid files are then provided via the web server. For each updater, appropriate batch and Check EXE files are provided.

Generation of Pattern Info Files

In case of a successful check, each provided Check EXE generates a Pattern Info INI which contains the current Pattern Information. These files can then be further processed by iQ.Suite Update Manager. On the one hand, all generated Pattern Info INI files are merged and listed in a central global Pattern Info INI which can be called via the web server. The global information can be provided also as HTML or JSON. On the other hand, the Pattern Information of the individual scanners can also be collected in CSV history files.

Automatic Reload of Updater Configurations

The automatic configuration reload can be activated and used for the scanner update processes. The configuration reload will check each scanner section in the configuration file for modifications. If changes are detected, the respective scanner update process will be restarted with updated settings. A service restart is only required when updating general settings then (e.g. server settings, logging).

Sending Success and Error Emails

In case of success and/or error with updates and when the Update Manager server is started, configurable email notifications can be sent to user-defined recipients.

Logging, Event Log and Access Log

iQ.Suite Update Manager writes a global processing log and for every updater a separate updater log. Additionally, Event log entries can be written in the Windows Event Viewer as well as on the file system. Access attempts to webserver resources can be logged into separate Access Log files.

Licensing: Using the Update functionality requires a license.

2 Installation

iQ.Suite Update Manager is installed with a separate setup file (EXE) and automatically registered as a Windows Service.

The setup file can be downloaded from the download area of the GBS website:
https://www.gbs.com/de/downloads/iqsuite/update-manager

Supported Operating Systems:

- Windows Server 2012 and 2012 R2 (each 64 Bit)
- Windows Server 2016 (64 Bit)
- Windows Server 2019 (64 Bit)

2.1 New Installation

To install iQ.Suite Update Manager, proceed as follows:

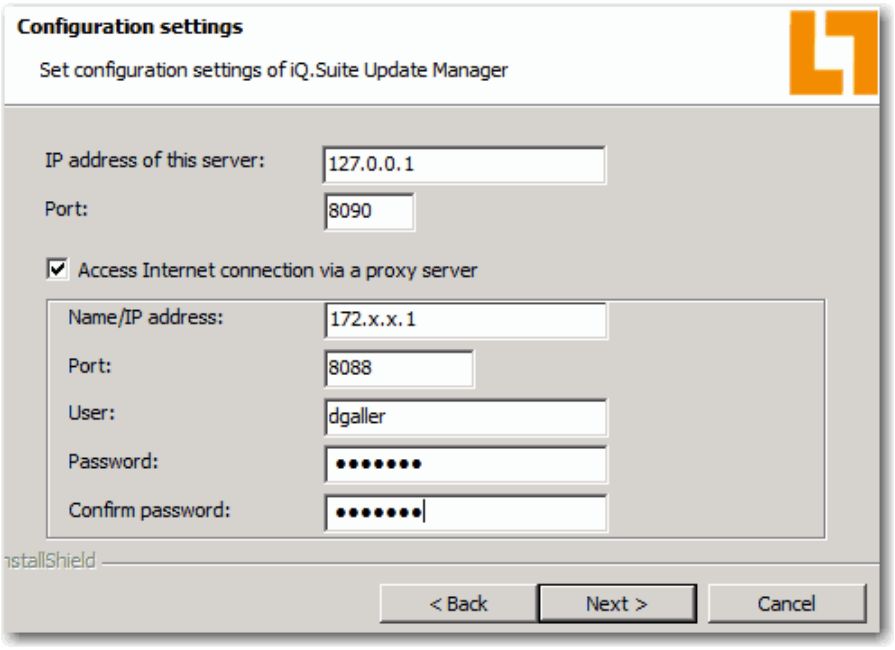
1. Run the setup: *iQ.SuiteUpdateManager.exe*.

   The Installation Wizard is started.

2. Select the installation folder:

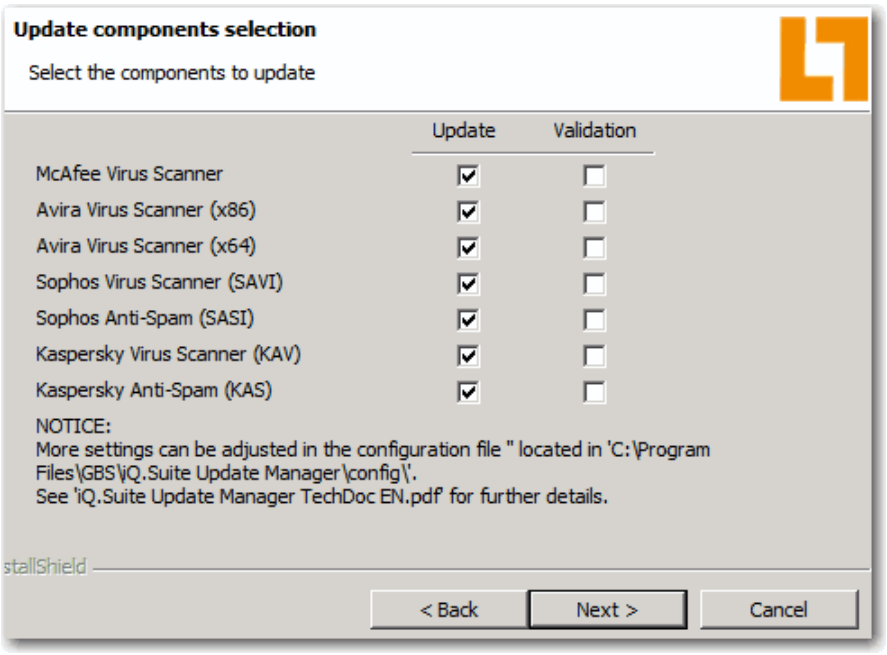
   Default: `C:\Program Files\GBS\iQ.Suite Update Manager\`

3. Configure the Update Manager server to be used to store the downloaded updates. Additionally, you can configure a proxy server if required:



- Update Manager Server:
  - **IP address of this server**: IP address of the Update Manager server.
  - **Port**: Port that can be used by the iQ.Suite to access the Update Manager server (default: 8090). In case you changed the defsult port, please make sure that the specified port is not used by another application such as iQ.Suite KeyManager.
- **Access Internet connection via a proxy server**: If a proxy server is required for Internet connections in your network environment, enable this option.
  - **Name/IP address**: Full name or IP Address of the proxy server, for example *proxy.mydomain.de* or *172.x.x.1*.
  - **Port**: Port that can be used by the Update Manager to access the proxy server.
  - **User** and **Password** (optional): Authentication information to be used by the Update Manager to login to the proxy server.

4. Click on NEXT and select under **Update** the components (OEM products) of which the updates are to be downloaded by using the Update Manager:

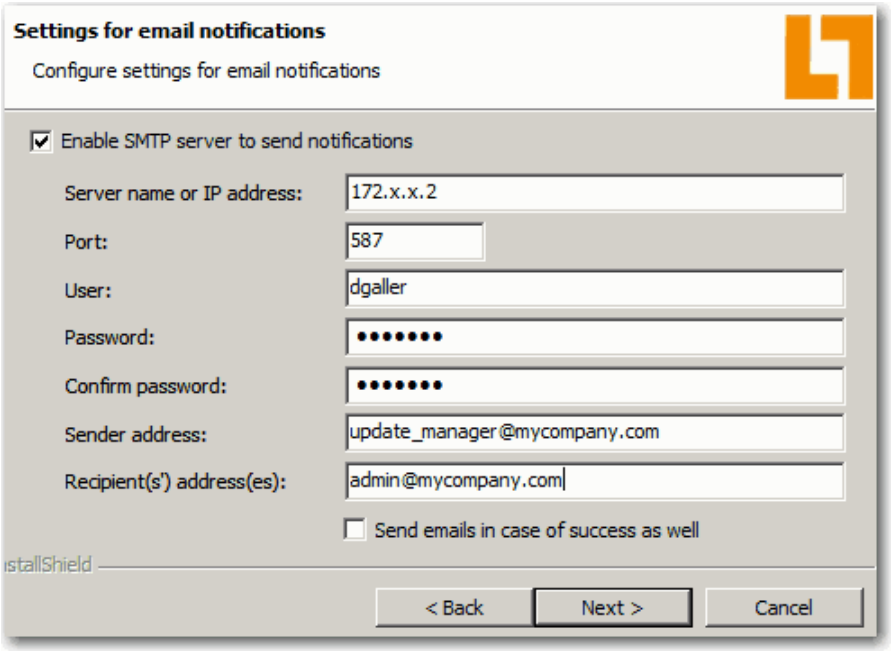


Update: Corresponds to the key `Enabled` in the `[<virus_scanner> Update]` sections.

After the download of updates, it is possible to check whether the downloaded files are consistent. For this, enable the **Validation** option for the desired products.

Validation: Corresponds to the key `CheckFiles.Enabled`.

5. Click on NEXT:



- **Enable SMTP server to send notifications**: Enable this option if you want notifications to be sent in case of update errors. Then, make the settings required for the SMTP server:
  - **Server name or IP address**: Server name or IP address of the SMTP server.
  - **Port**: Port to access the SMTP server.
  - **User**: User to authenticate against the SMTP server.
  - **Password** and **Confirm password**: Password of the user specified above.
  - **Sender address**: Specify which email address is to be used as the sender address. This can also be a fictive email address.
  - **Recipient(s') address(es)**: Email address(es) to which the notifications are to be sent.
- **Send emails in case of success as well**: Enable this option if you want notifications to be sent also in case of successful updates.

These settings correspond to the keys in the `[SMTPSettings]` section and to the first three keys under `[ErrorMailReporting]`/`[SuccessMailReporting]`.

6. Click on NEXT.

   By default, the iQ.Suite Update Manager Service will be automatically started during installation. If you do not want this, disable the corresponding option in the Setup dialog.

7. Click on INSTALL -> FINISH.

   The `files`, `logs` and `work` directories (default) are created and used in the installation directory when the 'iQ.Suite Update Manager' Service is running. The `files` directory contains the downloaded engine and pattern updates which are provided by the internal web server.

In the installation directory, you can also find the `config` directory. This directory contains the *UpdateManager.ini* file in which you can edit the initial configuration manually. Refer to Configuration of iQ.Suite Update Manager.

8. Copy your license file *UpdateManager.lic* to `...\iQ.Suite Update Manager\license`.
9. Restart the 'iQ.Suite Update Manager' Service manually so that the license file can be detected.

2.2 Update Installation

An **update installation from Version 6.0 to 7.0** of iQ.Suite Update Manager is possible by using the setup file.

If you want to update an older version, please read the notes contained in the techDoc of your product version.

3 Configuration

3.1 Configuration of iQ.Suite Update Manager

The configuration of the ‘iQ.Suite Update Manager’ Service is saved in the *UpdateManager.ini* configuration file that was installed by the setup.

This configuration file consists of sections. The sections `[Version]` and `[Server]` must exist. The other sections are optional and can be adjusted if required.

The sections contain several key value pairs that control the Update Managers behavior. Some few keys require reasonable values be configured. The remaining keys have default values. These are automatically used when a key is missing.

In order that adjustments in the configuration file take effect, you have to restart the Service.

Whenever the configuration file and the template file (*template.ini*) cannot be found when the Service is started, the Service is stopped with an error and an error log file *update_manager_startup_error_JJJJMMTT.log* is created. The template is used to create a new configuration file in case the configuration is missing.

Sections in the Configuration File:

- `[Version]`

  This section is used to match the configuration file with the Update Manager.

Important note: Do not make any manual changes here since otherwise the functioning of the service cannot be guaranteed.

- `[Server]`

The keys contained in this section control the behavior of the Update Manager server. These keys do not have any default values and correspondingly reasonable values need to be entered.

Key	Meaning
`IP`	IP address of the Update Manager server.
`Port`	Port to use for accessing the Update Manager server.
`DisplayConfig`	Enables/Disables displaying the configuration file under the URL `http://<ip>:<port>/config`
`ListDirectories`	Enables/Disables 'Directory Listing' of the web server.
`BaseFiles Directory`	Defines the directory on the file system from where the Update Manager server is to handle requests of the iQ.Suite client. Relative paths apply relative to the directory that contains the *UpdateManagerService.exe*. If the defined directory does not yet exist, it is created. Each downloader creates a separate sub-folder.
`BaseLog Directory`	Defines the base directory for log files. Relative paths apply relative to the directory that contains the *UpdateManagerService.exe*. If the defined directory does not yet exist, it is created.
`BaseWork Directory`	Defines the working directory which is required for CheckFiles, the use of a PreferINI as well as for some downloaders (Kaspersky).A separate subdirectory is created for each downloader. CheckFiles and PreferINI can be used simultaneously. If the working directory is also required by the downloader, an additional subdirectory `FILES` is created. The files in this subdirectory are then used for the CheckFiles feature or the Prefer.ini semantics. Relative paths apply relative to the directory that contains the *UpdateManagerService.exe*. If the specified directory does not exist yet, it will be created (in case CheckFiles is used).

- `[Config Reload]`

The keys contained in this section handle the automatic configuration reload due to changes in update sections.

Key	Meaning	Default
`Enabled`	Enables the configuration reload on configuration changes (1). The configuration file is recurringly checked for modifications. If changes are detected in an Update section, the respective updater will be restarted with the new settings (no service restart required). If disabled (0), the following parameters are not taken into account.	`0`
`FileCheck Interval`	Defines the interval at which the configuration file is checked for modifications (in seconds).	`60`

- `[Proxy]`

The keys contained in this section describe optional proxy server information that has to be taken into account when update files are downloaded.

Key	Meaning	Default
`Enabled`	Enables the use of the proxy server settings (1). If disabled (0), the following parameters are not taken into account.	`0`
`Address`	IP address of the proxy server	`<empty>`
`Port`	Port of the proxy server	`<empty>`
`User`	Username of the proxy server	`<empty>`
`Password`	The user's password	`<empty>`

- `[KAV Update]/[KAS Update]/[Key Update]/[McAfee Update]/ [SAVAPI Update]/[SAVAPI64 Update]/[SAVI Update]/[SASI Update]`

The keys contained in these sections describe the settings regarding the file downloads for the update of KAV, KAS, Key, McAfee, SAVAPI, SAVAPI64, SAVI and SASI.

Key	Meaning	Default
`Enabled`	Enables the file download for the updates (1). If disabled (0), the following parameters are not taken into account.	`1`
`UpdateUrl`	URL from where the downloads are taken. The up-to-date URLs for the corresponding modules are set initially: KAV: http://updater.gbs.com/kav/ KAS: http/updater.gbs.com/kas KEY: http://updater.gbs.com/keys/ McAfee: http://update.nai.com/Products/commonupdater/ SAVAPI / SAVAPI64: http://professional.avira-update.com/update http://professional.avira-update.net/update SAVI: http://updater.gbs.com/savi/ SASI: http://updater.gbs.com/sasi For each updater, multiple URLs can be separated by a semicolon. If necessary, the Update Manager tests out all existing URLs before an update is aborted with an error. This key must exist.	-
`SubDir`	Subdirectory relative to the main directory (`BaseFilesDirectory`) of the Update Manager server where the files are saved to. The key must exist.	-

`Platform`	Defines the platform(s) for which updates are to be provided. Possible values are:
	`[KAV, KAS]`
	`Linux86, Linux64, Win86, Win64, All`
	`[SAVAPI]`
	`Linux, LinuxComp, Windows, All`
	Separate each value by a semicolon.
	`[SAVAPI64]`
	`Linux, LinuxComp, Windows, All`
	`[SAVI]`
	`AIX, Linux86, Linux64, Win86, Win64, All`
	Separate each value by a semicolon.
		`All`

Key	Meaning	Default
	`[SASI]` `Linux, Windows,All`	
`VDFType [SAVAPI]`	Defines the virus definition file format to be downloaded: `vdf:` SAVAPI1, old vdf format (up to iQ.Suite 18.1 (IBM Domino) / 14.1 (Exchange/SMTP)) `xvdf_v1:` SAVAPI1, new xvdf-Format (as of iQ.Suite19.0 (IBM Domino) / 15.0 (Exchange/SMTP)) `xvdf_v2:` SAVAPI2 (as of iQ.Suite 18.1 (IBM Domino) / 14.1 (Exchange/SMTP), NO Solaris) `All:` All definitions mentioned above can be combined together. Separate each value by a semicolon.	`xvdf_v1; xvdf_v2`
`ScannerKeys [Key]`	Specifies the scanner license files to be downloaded and provided by the Key update. Possible values: `kav.key, kas.key, savapi.key, savapi2.key` Separate each value by a semicolon.	-
`Provide PreferINI`	Provides downloaded files indirectly via a *Prefer.ini* and subdirectory (`UseDir` and `UpdateDir`), instead of directly. Requires `BaseWorkDirectory`. This ensures that clients are provided with consistent updates and do not collide with the Update Manager download process. This key must be set for [SAVI] in order to enable the known update behavior! The option is enabled by default for [KAV] and [KAS] as well. **Only KAV**: For iQ.Suite clients < 15.0 (iQ.Suite for IBM Domino) or < 19.0 (iQ.Suite for Exchange/SMTP), the option must be disabled. Otherwise, the clients won't be able to retrieve updates.	`0`
`Update Interval`	Defines the interval at which updates are obtained (in minutes).	`60`
`Update Retry Interval`	Defines the interval at which updates are obtained when a previous update task was cancelled due to an error (in minutes).	`10`

Key	Meaning	Default
Cleanup Enabled	**[only McAfee, SAVI and SASI]** Enables the deletion of files that are not required (any longer) after each update.	1
WriteUpdate Report	Writes updater-specific Report logs after each update to the BaseLogDirectory.	1
CheckFiles. Enabled	Enables/Disables the CheckFiles mode. In case this mode is enabled, the downloaded files are first saved to the BaseWorkDirectory (Section [Server]). Afterwards, the provided Check EXE is executed on this directory per call of a Check batch. If this file returns a success, the files are copied from BaseWorkDirectory to the actual Update directory in the BaseFilesDirectory. Additionally, a Pattern Info INI is written. For [Key], there is no separate CheckFiles mode since downloaded license keys are tested in the course of other checks (KAV, KAS, SAVAPI)! Also refer to CheckFiles.BatchPath und CheckFiles.Timeout.	0
CheckFiles. BatchPath	Path to the executable check file which checks the files contained in the BaseWorkDirectory. Relative paths apply relative to the directory that contains the *UpdateManagerService.exe*. The given default batch, corresponding Check EXE and, if required, a scanner SDK are provided by iQ.Suite Update Manager. Also refer to CheckFiles.Enabled.	batches/ check_* .bat
CheckFiles. Timeout	Defines the timeout for calling the executable check file (in minutes). Also refer to CheckFiles.Enabled.	5

CheckFiles32.*	**[KAV, MCAFEE, SAVI]** For [KAV], [McAfee] and [SAVI], both 32 bit and 64 bit patterns can be downloaded and checked simultaneously. With the normal CheckFiles options, the 64 bit patterns are checked. With the additional CheckFiles32 options, the 32 bit patterns can be checked as well. The configuration of the CheckFiles32 options (Enabled, BatchPath, Timeout) is analogous to the CheckFiles options. Default BatchPath: [KAV]: batches/check_kav_32.bat [McAfee]: batches/check_mcafee_32.bat [SAVI]: batches/check_savi_32.bat	...

- [KAV Update]/[KAS Update]/[SAVAPI Update]/[SAVAPI64 Update]

For updating KAV, KAS, SAVAPI and SAVAPI64, specific keys are additionally needed since updating is performed by an additional, external program.

Key	Meaning	Default
UpdateExe	Path to the external program which is used to perform the update. Relative paths apply relative to the directory that contains the *UpdateManagerService.exe*. Default path SAVAPI: *savapi/avupdate.exe* Default path SAVAPI64: *savapi64/avupdate.exe* Default path KAV/KAS: *kasperksy64/KasperskyUpdate.exe* This key must exist.	-
Update Timeout	Defines the timeout for calling the external program (in minutes).	5

[Global Pattern Information]

The keys specified in this section regulate the creation of the global Pattern Info INI and other options which are based on it. The global Pattern Info INI contains the Pattern information relative to all updaters for which CheckFiles is enabled and a local Pattern Info INI was written. According to this, no global Pattern Info INI can be generated if CheckFiles is disabled for all updaters.

Key	Meaning	Default
Enabled	Enables the generation of the global Pattern Info INI (1). When disabled (0), the following parameters will be ignored.	1
INIFile	Name of the global Pattern Info INI. A relative path applies relative to the BaseFilesDirectory, i.e. the file can be accessed via the web server by default.	patterns.info
GenerateHTML	Enables the generation of an HTML file from the global Pattern Info INI and an HTML template (1).	1
HTMLTemplate	Path to the HTML template for the generation of the global Pattern Info HTML. A relative path applies relative to the config directory. The HTML template may contain the special HTML tag <for-each-upd>. If so, the HTML code framed with this tag is set for every active updater. In the framed HTML code, some wildcards can be used. These wildcards are resolved accordingly during the HTML generation: {{updater}} -> Name of the updater {{engine}} -> Version of the Engine {{pattern}} -> Version of the patterns {{lastupdate}} -> Date of the last successful updates	global_pattern_info_template.html
HTMLFile	Name of the generated HTML file. A relative path applies relative to the BaseFilesDirectory, i.e. the file can be accessed via the web server by default.	pattern_info.html

GenerateJSON	Enables the generation of a JSON string from the global Pattern Info INI (1).	1
JSONUrl	Sub URL for accessing the JSON string. The generated JSON is created and served from memory; no file is created.	/pattern_info.json
GenerateHistCSV	Enables the generation of CSV history files (1). When enabled, the Pattern Info files are saved for each updater with active CheckFiles to separate CSV history files. These files contain the Pattern information for the last (max. 1000) updates in CSV format, sorted from new to old. Here, such updates for which the version strings of the Engine and patterns were not changed are ignored.	1
HistCSVSuffix	File extension for the CSV history files, e.g.: mcafee_pattern_info.hist.csv The CSV history files are saved to the BaseLogDirectory.	_pattern_info_hist.csv

[Logging]

The keys contained in this section control the logging for the iQ.Suite Update Manager.

Key	Meaning	Default
Enabled	Enables logging (1). If disabled (0), the following parameters are not taken into account.	1
LogLevel	Sets the log level for the logging. Possible log levels: Error, Notice, Info, Detail, Debug, Noisy Use the lowest log level Error to log errors only. The higher the log level, the more information is logged.	Detail
LogDir	Directory to which the log files are saved. Relative paths apply relative to the BaseLogDirectory. If the specified directory does not yet exist, it is created.	<empty>
LogFile Prefix	Prefix for the filenames of the log files.	update_ manager
LogFile Postfix	File type to be used for the log files.	.log
LogFile Interval	Defines the interval at which a new log file is written. Possible values are: Monthly, Weekly, Daily, Hourly	Daily
LogFile Count	Defines the maximum number of different log files that are kept.	14

[EventLog]

The keys contained in this section control the event logging for the iQ.Suite Update Manager.

Key	Meaning	Default
Enabled	Enables Event logging (1). If disabled (0), the following parameters are not taken into account.	1
LogLevel	Sets the Log level for the Event logging. Possible log levels: None, Min, Med, Max, All With log level None, no event logging is performed. The higher the log level the more information is logged.	Med
LogToFile	Is responsible for the event logging is written to a file instead of to the Windows Event log (1). If not used (0), the following parameters are not taken into account.	0
LogDir	Directory to which the event log files are saved. Relative paths apply relative to the BaseLogDirectory. If the defined directory does not yet exist, it is created.	<empty>
LogFile Prefix	Prefix for the filenames of the event log files.	update_ manager _events
LogFile Postfix	File type to be used for the event log files.	.log
LogFile Interval	Defines the interval at which a new event log file is written. Possible values are: Monthly, Weekly, Daily, Hourly	Daily
LogFile Count	Defines the maximum number of different event log files that are kept.	14

[AccessLog]

The keys contained in this section control the webserver access logging.

Key	Meaning	Default
Enabled	Enables access logging (1). If disabled (0), the following parameters are not taken into account	1
LogDir	Directory to which the access log files are saved. Relative paths apply relative to the BaseLogDirectory. If the defined directory does not yet exist, it is created.	<empty>
LogFile Prefix	Prefix for the filenames of the access log files.	update_ manager _access
LogFile Postfix	File type to be used for the access log files.	.log
LogFile Interval	Defines the interval at which a new access log file is written. Possible values are: Monthly, Weekly, Daily, Hourly	Daily
LogFile Count	Defines the maximum number of different access log files that are kept.	14
Entry Template	Specifies the formatting of the access log entries. For each access attempt on the webserver, a log line will be constructed using the given template string. The following wildcards can be used: {timestamp} -> Date and time of the request {remote_ip} -> IP of the requesting client {user_agent} -> UserAgent string of the requesting client {uri_requested} -> Requested ressource on the webserver {response_code} -> HTTP response code Default: {timestamp} | {remote_ip} | {user_agent} | {uri_requested} | {response_code}	

- `[SMTPSettings]`

  Use these sections to configure the SMTP settings to enable the Update Manager to send emails (refer to the sections `[ErrorMailReporting]` and `[SuccessMailReporting]`).

Key	Meaning	Default
`SMTPHost`	Address of the SMTP host for sending emails. This key must exist.	-
`SMTPPort`	SMTP port for sending emails. This key must exist.	-
`SMTPAuth Username`	SMTP username with which to authenticate when emails are sent. If no username is specified for this key, no authentication is performed (default).	`<empty>`
`SMTPAuth Password`	Password of the SMTP username with which to authenticate when emails are sent.	`<empty>`
`SMTPUseSSL`	Enables SMTPS/SSL encryption.	`0`
`SMTPCert Path`	Specifies the CA certificate to be used to check the received server certificate (during the SSL handshake). If no certificate is specified, the server certificate is trusted without check.	`<empty>`

- `[ErrorMailReporting]/[SuccessMailReporting]`

  Use these sections to configure the sending of emails in case of erroneous or successful updates. Both Reporting options require correct SMTP settings (refer to the `[SMTPSettings]` section).

  The settings in these sections apply *globally* to all updaters. However, they can be overwritten locally if required (refer to Overwriting parameters locally).

Key	Meaning	Default
`Enabled`	Enables the sending of error/success emails for all updaters (`1`). When disabled (`0`), the parameters mentioned below are not considered.	`0`
`SMTPSender`	Sender address of the success/error email. This key must exist.	-

SMTPRecipient	Recipient address(es) of the error/success email. In case of multiple recipients, separate each entry by a semicolon. This key must exist.	-
MailSubject	Subject of the error/success email. Possible wildcards: {module}: updater name {error_code}: error code string {error}: short error description {date}: Date and time in local time {date_utc}: Date and time in UTC time {last_update}: Date and time of the last successful update (in local time). {last_update_utc}: Date and time of the last successful update (in UTC time). This key must exist.	-
MailBodyFile	Path to a file whose content is to be used as email content. Relative paths apply relative to the configuration file. Possible wildcards: refer to MailSubject and {report} This key must exist.	-
ExcludedErrorCodes	Specifies error code strings for which no error mails shall be sent. Has no impact on success mails. Wildcards '*' and '?' can be used. Separate multiple values by a semicolon.	<empty>
Threshold	Defines a threshold for the mailing (in minutes). **In case of error emails**, sending emails can be postponed: Only when the time since the last successful update exceeds the specified threshold, an error email is sent. **In case of success emails**, sending emails is disabled when the threshold time expires, provided that no update error occurred in the meantime. If this key is not set or empty, no threshold is used and emails are always sent.	<empty>

The contents of the TXT files in the `config` directory are inserted into the message body of the error/success emails. If required, you can edit these files or create and use new files, provided that these files are referenced accordingly.

Overwriting parameters locally

The global mailing settings described above can be overwritten for each updater separately (local settings). For this, add in the updater's section new key value pairs as follows:

```
ErrorMailReporting.{SCHLÜSSEL}={LOKALER WERT}          or
SuccessMailReporting.{SCHLÜSSEL}={LOKALER WERT}
```

Example:

For all updaters, an email is to be sent in case of errors (global settings), except for the SASI update (local setting):

```
[ErrorMailReporting]      Enabled=1    (. . .)
[SASI Update]             (. . .)      ErrorMailReporting.Enabled=0
```

In the `[Server]` section, you can overwrite the mailing settings locally as well. In this case, emails will be sent in case of errors or success also when the 'iQ.Suite Update Manager' Service is started.

In contrast to the updater sections, the ‚Enabled' keys has to be explicitly set in the `[Server]` section, not depending on the value in the `MailReporting` sections.

3.2 Configuration of the Engines in iQ.Suite

The Engines obtain their updates from the Update Manager server. Therefore, the Engines must be configured accordingly. For this, the URL to the **Download source** must be specified.

The download source is the Internet address that is used by the Update Manager to provide the downloaded update files. The download source consists of the IP address and the port of the Update Manager server and the name of the directory that contains the updates (`SubDir`):

```
http://<IP address>:<Port>/<SubDir>.
```

Take these entries from the *UpdateManager.ini*. Example:

```
[Server]
IP=127.0.0.1
Port=8090

[McAfee Update]
SubDir=mcafee
```

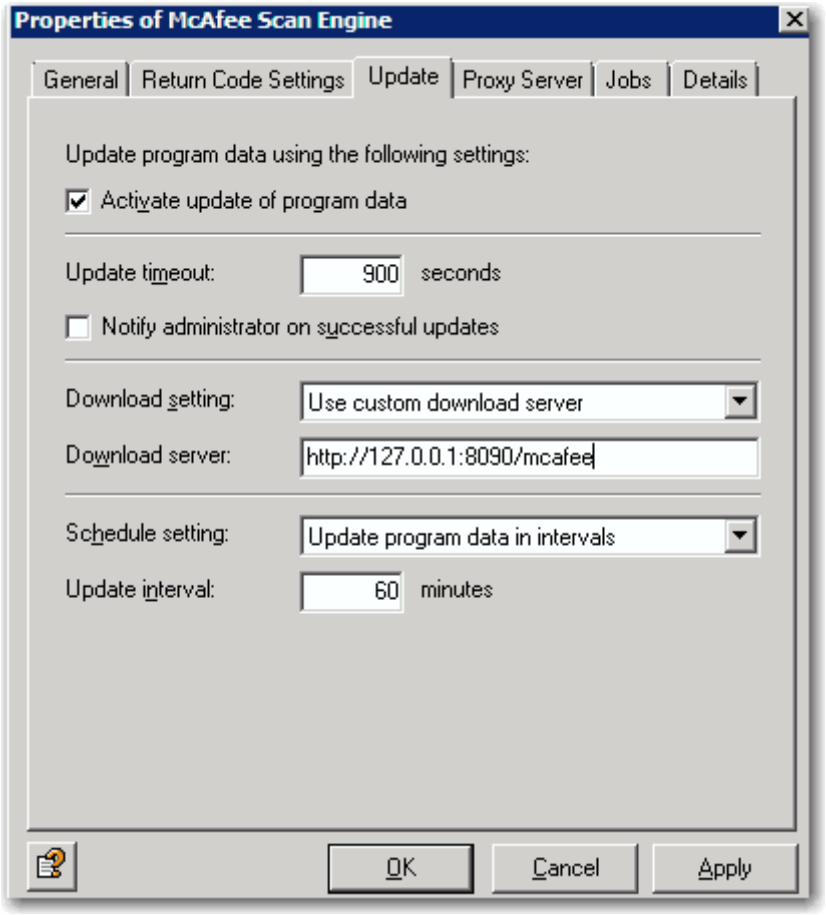
=> Download source: `http://127.0.0.1:8090/mcafee`

The following sections describe the settings required to connect the Scan Engines to the Update Manager. For further information on the Engine configuration, please refer to the iQ.Suite-Administration Manual. Download on www.gbs.com.

3.2.1 iQ.Suite Microsoft

For any Engine, proceed as follows:

1. Open the Engine.
2. In the **Update** tab, enter the URL to the **download server**:



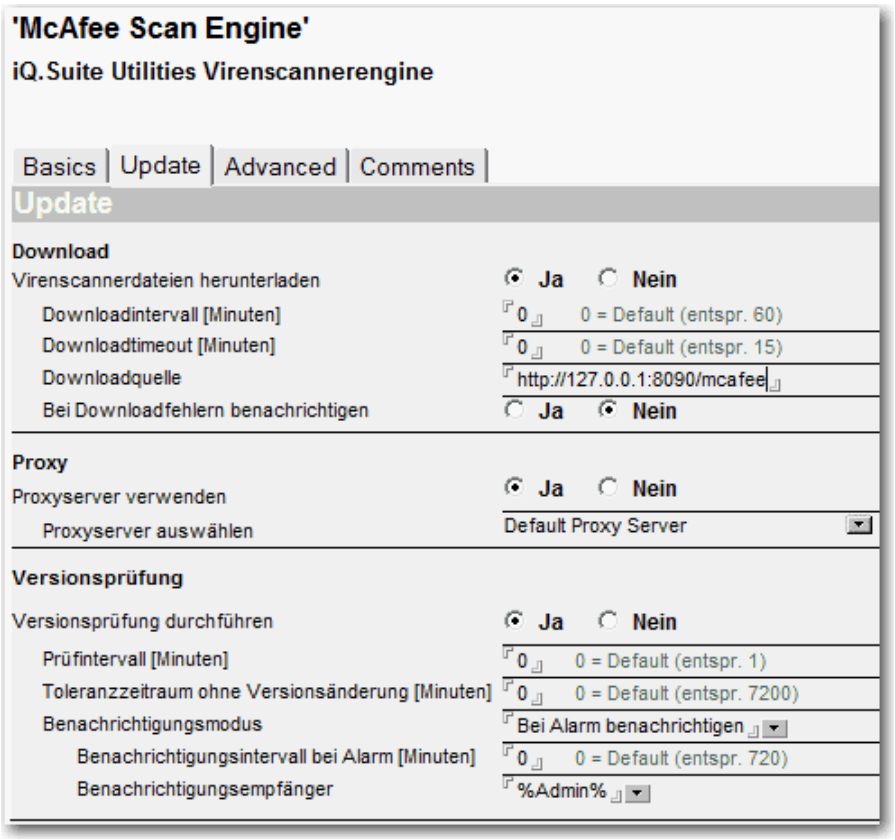
3. Click OK and save the configuration.

3.2.2 iQ.Suite Domino

3.2.2.1 All Engines, Except Avira Scan Engine (SAVAPI1)

The following settings are valid for all virus scanners, except **Avira Scan Engine (without APC Option)**. For SAVAPI, please refer to section 3.2.2.2.

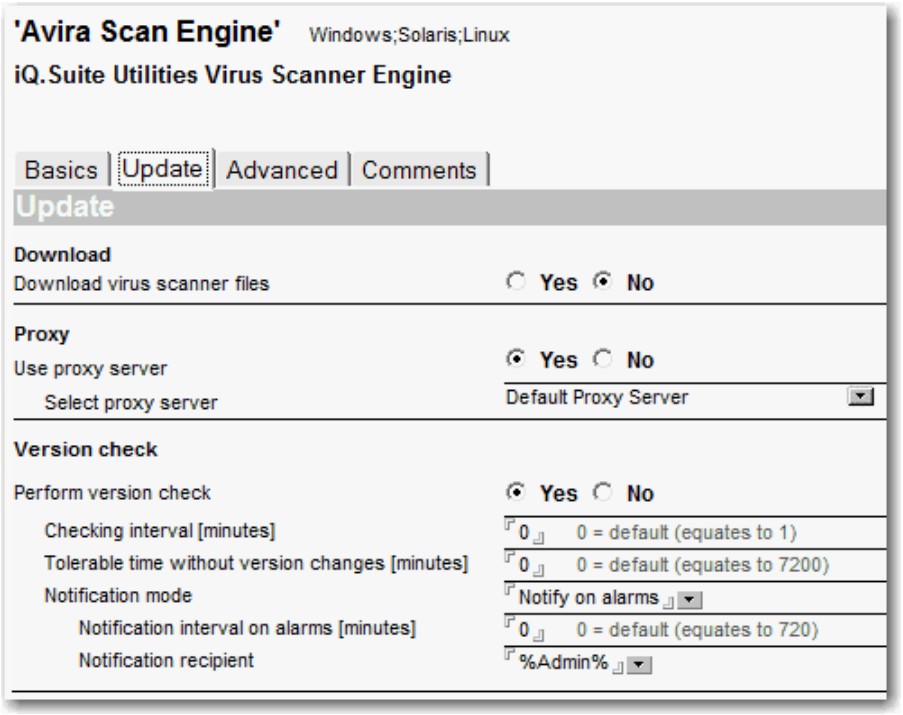
1. Open the Engine configuration document.
2. Select under DOWNLOAD -> DOWNLOAD VIRUS SCANNER FILES the option 'Yes' and enter the URL in the **Download source** field:



3. Save the configuration.

3.2.2.2 Only Avira Scan Engine (SAVAPI1)

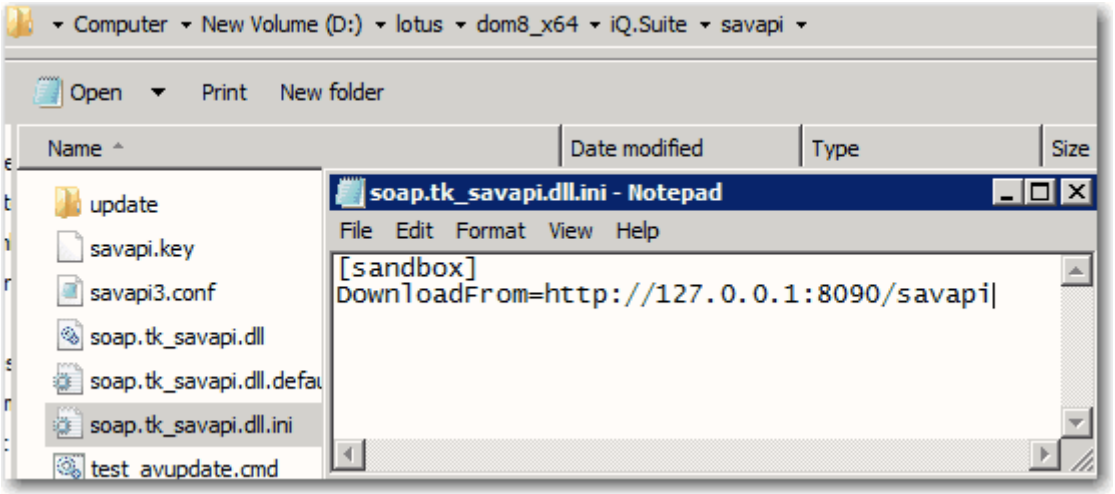
1. Open the configuration document of the Avira Scan Engine.
2. In the **Update** tab, select under **Download virus scanner files** the option 'No':



3. Save the document.
4. Under `<Domino directory>\iqsuite\savapi` open the file **soap.tk_savapi.dll.ini** and add in this file the following line:

   `DownloadFrom=http://<IP-Adresse>:<Port>/<SubDir>`

   Example:
   `DownloadFrom=http://127.0.0.1:8090/savapi`



5. Save the file.
6. For your settings to apply, restart the `tm_grab` task.

4 About GBS

GBS Europa GmbH is a leading vendor of solutions and services in the fields of messaging security and workflow for the Domino 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.

Web site: www.gbs.com

Email address: info@gbs.com

Locations: www.gbs.com/en/locations

GBS
A BULPROS COMPANY