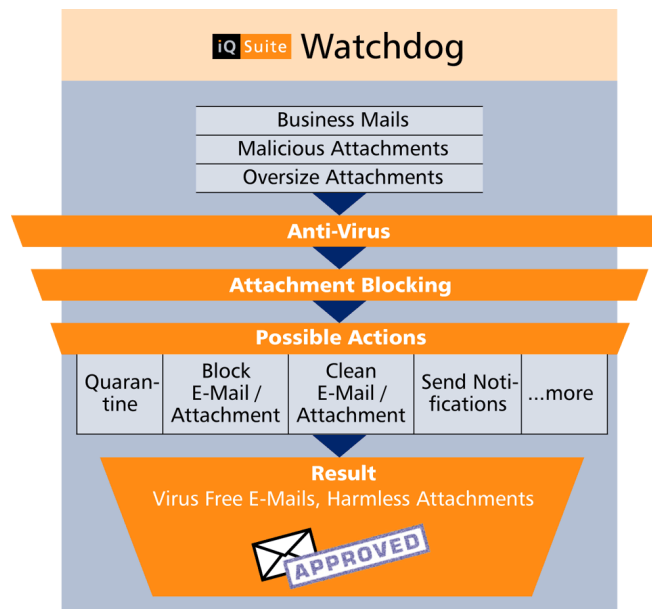


iQ.Suite Watchdog

– Built-In Virus Scan Engine –



iQ.Suite 5 for Exchange/SMTP

Contents

1 General.....	3
1.1 Components	3
1.2 System Requirements	3
1.3 Licensing	4
1.4 General Features.....	4
1.5 Dialer Support	4
1.6 Updates	4
2 Technical Concept.....	5
3 Technical Implementation.....	6
3.1 DLL Load / Service Initialization	6
3.2 Communications.....	6
3.3 Scheduler	6
3.4 Configuring the Service	7
3.5 Update Service	7
3.6 Log file.....	7
4 Commissioning the Virus Scanner	7
4.1 Configuration Procedure.....	7
4.2 Known Issues	8
5 Appendix A – Structure of SAVAPI.INI	9
5.1 Possible Entries in SAVAPI.INI	9
5.1.1 Port Number	9
5.1.2 Number of Simultaneous Connection Requests	9
5.1.3 Directory for Temporary Files	9
5.1.4 Directory for Updates	10
5.1.5 Engine File Names	10
5.1.6 Searching the Archive.....	10

5.1.7	Maximum Recursion Depth in Archive Searches.....	10
5.1.8	Automatic Archive Recognition	10
5.1.9	Server Name for Updates	11
5.1.10	Using a Proxy Server for Updates	11
5.1.11	Proxy Server Address.....	11
5.1.12	Proxy Port Address.....	11
5.1.13	User Name for Proxy Server (Proxy Authentication).....	11
5.1.14	Password for Proxy Server (Proxy Authentication).....	11
5.1.15	Search Interval for New Updates	12
5.1.16	Download Control Log file Name	12
5.1.17	Engine Log file Name.....	12
5.1.18	Maximum Size of Engine Log file.....	12
5.1.19	Notification of AntiVir Guard.....	12
5.1.20	Dialer Detection	13
5.1.21	Macro Virus Heuristics	13
5.1.22	Win32 File Heuristic.....	13
6	Appendix B – Structure of SAVAPIDL.INI	14
6.1	Possible Entries in SAVAPIDL.INI	14
6.1.1	Port Number	14
7	About GROUP Technologies AG	15

1 General

From version 4, iQ.Suite Watchdog includes the AntiVir virus scan engine powered by H+BEDV, which (the SAVAPI) is automatically installed when you reinstall iQ.Suite or run a program update.

Definition of names

- SAVAPI is the technical implementation of the H+BEDV AntiVir product.
- In the documentation and the setup program and configuration, the antivirus product is referred to as AntiVir.

AntiVir can be activated both manually from within iQ.Suite and with an entry in the license file.

1.1 Components

The SAVAPI 2, included from version 4, consists of the following components:

- SAVAPI.DLL – the interface to the application.
Passes all calls on to the scan engine.
- The search and repair engine
The part of the program that performs the actual virus scan. This part is implemented as a Windows SAVAPI service.
- A Windows service – the SAVAPI service – with built-in scheduler for handling incoming calls.
- An update service which downloads engine and VDF (Virus Definition File) updates through the intranet or Internet and installs them.

1.2 System Requirements

SAVAPI 2 has the following system requirements:

- Windows NT Server with Service Pack 6a and Microsoft Windows Installer 1.1
- Windows 2000 Server (SP2 recommended)
- Windows 2000 Advanced Server (SP2 recommended)
- Windows Server 2003
- At least 15 MB free disk space
- At least 6 MB available memory

1.3 Licensing

To use the scan engine within iQ.Suite, you need to acquire a separate license. Contact your sales partner for further information.

1.4 General Features

SAVAPI 2 features the following functions and changes:

- Fully-automatic engine and virus signature updates every 2 hours (preset) through the Internet.
- Optional support for the AntiVir Update Managers for central updates.
- Optional download of updates through a proxy server.
- Proxy authentication support.
- Maximum performance at lowest possible memory usage.
- Same scan and repair performance as all AntiVir products.
- Optional, recursive virus scanning in archives. The following archive formats are currently supported: ZIP, ZIP-Sfx, ARJ, ARJ-Sfx, TAR, GZ, ZOO, UUEncode/XXEncode, TNEF, MIME, BinHex, MSCompress, MS CAB, LZH/LHA, LZH/LHA Sfx, RAR, RAR-Sfx, JAR, BZ2, ACE, ACESfx. Support for additional formats will be available soon.
- Optimum scaling on multi-processor machines through variable number of worker threads.

1.5 Dialer Support

Although dialer support for SAVAPI is possible in principle, the SAVAPI library itself is not capable of issuing explicit messages when it detects a dialer and can not, therefore, influence the messages issued by iQ.Suite. Dialers therefore continue to be regarded as viruses. Disabled by default, dialer detection can be activated with an entry in SAVAPI.INI. For details about the corresponding entry, *ReportDialer*, see Appendix A.

1.6 Updates

By default, the built-in Internet update service automatically checks for engine and VDF updates every two hours and immediately installs any new updates. The new signatures are installed “on the fly”, so that the services do not have to be restarted, even when system utilization is at 100 %. The update interval can be changed with the *UpdateInterval* setting in SAVAPI.INI.

Updates can also be installed via CIFS Share or HTTP from another server, allowing the download control to work together with the AntiVir Internet Update Manager.

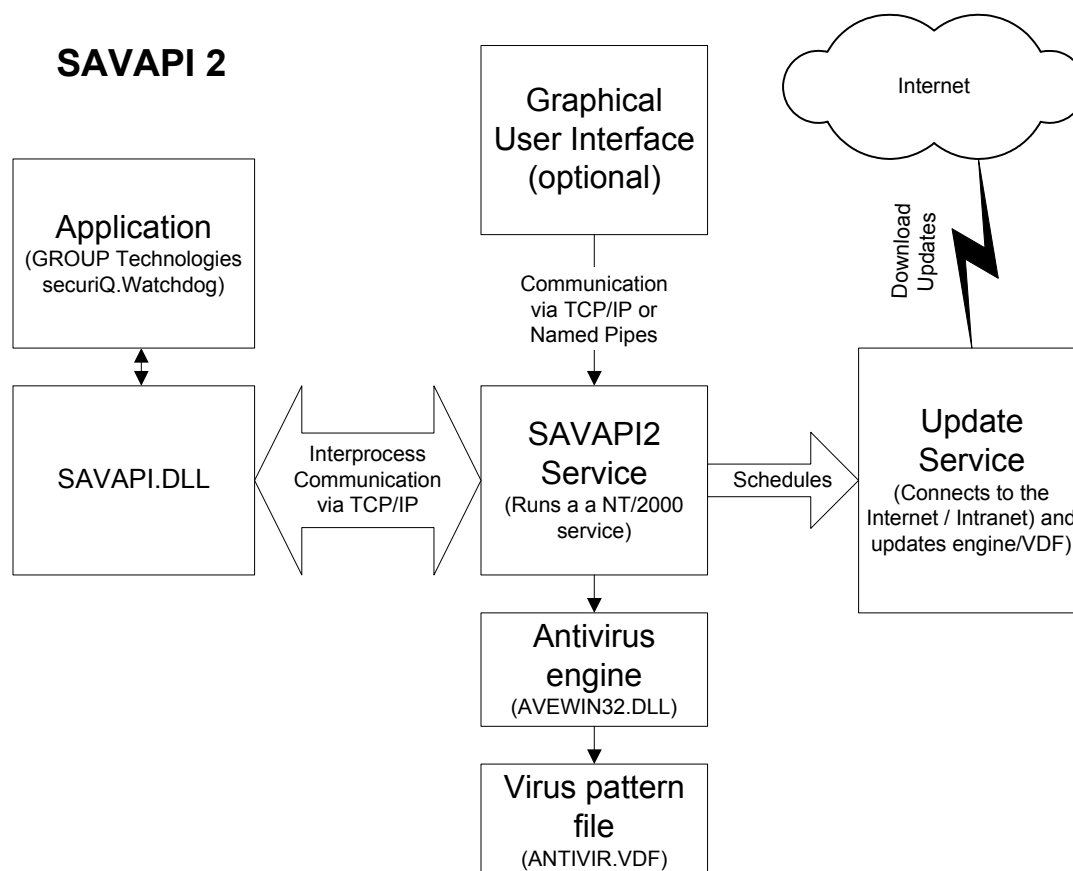
An update of the SAVAPI files themselves is not possible at present; this functionality is currently under development.

If security requirements prevent a direct update from the messaging server or if a central update server is used, the update process can be centralized with the H+B AntiVir Internet Update Manager.

The AntiVir Internet Update Manager is available free at <http://www.antiVir.de/dateien/antiVir/release/updman.exe>.

2 Technical Concept

To solve problems with updates that occurred with SAVAPI, H+B devised a new concept, which uses the same application interface as the old SAVAPI 1, but can update itself through a scheduler:



SAVAPI 2 consists of two applications:

- savapi2s.exe, which scans files for viruses
- savapi2r.dll, which implements the interface with iQ.Suite

Both applications communicate with each other through TCP connections. The TCP/IP protocol must therefore be installed on the computer.

3 Technical Implementation

3.1 DLL Load / Service Initialization

SAVAPI 2 DLL	SAVAPI 2 Service
<p>To provide support for multithreaded connections with the service, the DLL saves the thread parameters (stack, thread local memory) locally.</p> <p>Only the port number of the TCP server running the service, the engine and VDF names, scheduler data, etc. are saved globally.</p>	<p>When it is started, the service verifies the validity of all components (engine, and VDF) and waits for a connection with SAVAPI2.DLL.</p> <p>A port number for connecting to the DLL must be selected.</p> <p>If a component is invalid, the service continues to run but issues an error message.</p>

Further actions:

- Loading and initialization of the current engine and VDF
- Start of the scheduler
- Start of the log file

3.2 Communications

For communication, the proven client-server method is used.

- Internal communication is unencrypted.
- The service must only accept connection requests from localhost (127.0.0.1).
- Communications and all handling are secure for multithreading and multiprocessing.

3.3 Scheduler

SAVAPI 2 features a built-in scheduler, which can initiate timed updates.

3.4 Configuring the Service

The service is configured only through an INI file. If the INI file is missing, the scheduler uses the default update interval of two hours (random +/- 20 minutes). The INI file is saved when the service is terminated.

3.5 Update Service

The existing update service of AntiVir/2ks and the exact same procedure are used for the update: all active threads are stopped and deleted, Engine and VDFs are updated, and the threads are re-initialized and enabled again. The update itself is transparent for iQ.Suite.

3.6 Log file

The AntiVir service must write a central log file with the following structure:

DD-MM-YYYY,HH.MM.SS.MS [CATEGORY] Content

Categories:

INIT, EXIT, SCHEDULER, UPDATE

Content:

- Start message
- SAVAPI service, engine, and VDF versions
- Update service start
- Update service log
- Update service success/failure
- SAVAPI init call (license information)

4 Commissioning the Virus Scanner

4.1 Configuration Procedure

Having installed iQ.Suite, the “AntiVir Engine powered by H-BEDV (scan engine)” document under menu item **Scan Engine** is active.

If a virus scan job is also activated, for example “Virus Scanning With AntiVir Engine”, for e-mail, all inbound, outbound and internal mail is virus-scanned.

By default, the automatic update service runs every two hours. If new patterns or a new scan engine are available on the download server, it performs an update without affecting the function of iQ.Suite.

To set a shorter interval or if iQ.Suite and the download server are connected via a proxy server, you will have to edit the SAVAPI.INI file as follows:

1. Terminate the iQ.Suite Services.
2. In the Windows Computer Management under Services, terminate the SAVAPI service.
3. Open the SAVAPI.INI file, which is located in the AntiVir directory and which contains all relevant configuration settings.
4. In the Windows Computer Management, restart the SAVAPI service.
5. Start the iQ.Suite Services.

For further information, including the individual INI settings, see Appendix A.

If you have made any changes in SAVAPI.INI, especially regarding the proxy server, it is advisable to initially choose a short update interval. To see whether an update was successful, check the SAVAPI.LOG log file.

Carry out the above steps 1 to 5 every time you edit SAVAPI.INI.

4.2 Known Issues

SAVAPI does not contain a built-in decompressor.

The **ScanArchives** setting in SAVAPI.INI provides limited archive scanning functionality: ZIP files can be scanned, for example, but not RAR archives. The Parameter **ScanArchives** parameter must have the value 1.

To allow detection of viruses in all archives, the following must be true:

- The option “Extra archive scan with iQ.Suite unpacker” (Tab “Actions”) in the Virus Scan job must be enabled.

5 Appendix A – Structure of SAVAPI.INI

The AntiVir engine can be configured with the SAVAPI.INI initialization file. **Caution:** Normally, it is not necessary to change the engine configuration settings: the default settings are suitable for most situations.

When you first start the AntiVir engine, the SAVAPI service starts with safe default values and automatically creates the INI file in the AntiVir directory. Settings are written to the INI file only when the SAVAPI service is terminated for the first time.

Before you edit SAVAPI.INI, you must close the AntiVir engine (i.e. the SAVAPI service). To do this, start the Windows Computer Management application (**Start | Settings | Control Panel | Computer Management**), select Services in the left pane and terminate the entry “SAVAPI Service”. You can now edit SAVAPI.INI with Notepad. Having made your changes, restart the SAVAPI service and service that uses the SAVAPI.DLL.

Caution: If you are using a proxy server, adapt the SAVAPI.INI file for the online virus pattern updates by setting the parameters described from [Using a Proxy Server](#) for Updates.

5.1 Possible Entries in SAVAPI.INI

5.1.1 Port Number

This is the number of the TCP/IP port used for communications between the SAVAPI service and the SAVAPI.DLL. If this port is already assigned, you can change this value. Note, in this case, that you must also change the corresponding entry in SAVAPIDL.INI.

Example: PortNumber=18370

5.1.2 Number of Simultaneous Connection Requests

This value specifies the maximum number of connection requests that can be processed at the same time. For the AntiVir engine, the default value of 4 is sufficient. If you enter the value 2147483647 (SOMAXCONN), Windows uses the internal TCP-specific maximum value.

Example: MaxPendingConnections=4

5.1.3 Directory for Temporary Files

This value specifies the directory to which the engine writes its temporary files. By default, this is the subdirectory \TEMP of the installation directory. You can enter any other directory here, on any drive with sufficient free disk space.

Example: TempDirectory=C:\Program Files\GROUP Technologies\iQ.Suite\Engine\temp\

5.1.4 Directory for Updates

This is the working folder of the update service, to which the engine temporarily saves any updates downloaded from the Internet. Do not change this setting. The engine (i.e. the SAVAPI service) must have write-access to this directory.

Example: UpdateDirectory=C:\Program Files\GROUP Technologies\iQ.Suite\Engine\update\

5.1.5 Engine File Names

By default, the engine looks for its own files in the \ENGINE subdirectory of the installation directory. You can change these entries if necessary.

Example:

- VdfFileName=C:\Program Files\GROUP Technologies\iQ.Suite\Engine\antivir.vdf
- DllFileName=C:\Program Files\GROUP Technologies\iQ.Suite\Engine\avewin32.dll
- DatFileName=C:\Program Files\GROUP Technologies\iQ.Suite\Engine\savapi.dat
- ArchiveFileName=C:\Program Files\GROUP Technologies\iQ.Suite\Engine\avpack32.dll
- MsgDllFileName=C:\Program Files\GROUP Technologies\iQ.Suite\Engine\savapi2r.dll

5.1.6 Searching the Archive

If this parameter is set to 1, the engine also scans archives for viruses and other malware. The default is 0 (disabled).

Example: ScanArchives=0

5.1.7 Maximum Recursion Depth in Archive Searches

This setting lets you specify the maximum nesting depth within archives to which the engine scans for viruses. The default value of 2 should be sufficient in most cases. This value applies only if archive scanning has been activated with the ScanArchives parameter.

Example: ArchiveMaxRecursion=2

This value should be smaller than or equal to the setting in iQ.Suite Servers → Properties → General tab: "Expand nested archive file to the level of:". For further information, see section 5.3 of the administration manual.

5.1.8 Automatic Archive Recognition

Archives can be recognized in two ways: through their file extension and through their contents. Content detection (known as "smart detection") is the more reliable method, but takes longer. If the *ArchiveSmartDetection* option is enabled (set to 1), the engine tries to detect archives based on

their content; otherwise it uses the file extension. This value applies only if archive scanning has been activated with the ScanArchives parameter.

Example: ArchiveSmartDetection=1

5.1.9 Server Name for Updates

This is the URL at which the engine looks for updates (new virus signatures). If another server is used (for example by the AntiVir Update Manager), enter the corresponding URL here.

Example: UpdateUrl=http://www.antivir.de

5.1.10 Using a Proxy Server for Updates

If this value is enabled (1), the engine tries to download the updates through the specified proxy. By default, no proxy server is used.

Example: ProxyEnabled=0

5.1.11 Proxy Server Address

Here, you can enter the full name or IP address of the proxy server used for the update. This value is used only when “ProxyEnabled” is set to “1”.

Example: ProxyUrl=proxy.mydomain.com

5.1.12 Proxy Port Address

This parameter is used to set the port for the communication with the proxy server used for downloads, i.e. the port number of the proxy server. This value is used only when “ProxyEnabled” is set to “1”. Enter the proxy server’s port number here.

Example: ProxyPort=3128

5.1.13 User Name for Proxy Server (Proxy Authentication)

This parameter is used to set the user name to be used by the update service when connecting to the proxy server. This value is used only when “ProxyEnabled” is set to “1”.

Example: ProxyUserName=fsmith

5.1.14 Password for Proxy Server (Proxy Authentication)

This parameter is used to set the password to be used by the update service together with the user name when connecting to the proxy server. This value is used only when “ProxyEnabled” is set to “1”.

Example: ProxyPassword=password

5.1.15 Search Interval for New Updates

This parameter is used to set the time after which the update service is to check for new versions on the server specified under UpdateURL. The interval is specified in minutes. The default value is 120 minutes (2 hours). An automatic update of the engine and virus signatures is automatically performed immediately after the first action (virus scan). If this value is zero, automatic updating is disabled.

Example: UpdateInterval=120

5.1.16 Download Control Log file Name

This value specifies the name of the download control log file, which is written to the engine's installation directory. Note that the SAVAPI update service must have write-access to this directory.

Example: DlwLogFileName=DWLDSVC.LOG

5.1.17 Engine Log file Name

This value specifies the name of the engine log file, which can be located anywhere on the hard disk. Note, however, that the SAVAPI service must have write-access to this file. By default, the log file is called SAVAPI.LOG and is saved to the \ENGINE subdirectory of the engine's installation directory.

Example: LogFileName=C:\Program Files\H+BEDV\AntiVir SAVAPI2\Engine\savapi.log

5.1.18 Maximum Size of Engine Log file

This value specifies the maximum size in KB of the log file. If this size is exceeded, the oldest entries are automatically deleted. If set to 0, size limitation is disabled.

Example: LogFileSize=100

5.1.19 Notification of AntiVir Guard

Before the engine and AntiVir Guard or AntiVir for Windows Server can run on a machine, the SAVAPI must be notified to AntiVir Guard. If this does not happen, AVGuard handles any infected files before the SAVAPI (for example deleting them), so that the SAVAPI service may no longer be able to find the file to be scanned. If AntiVir Guard is not installed on the same computer, this option can be disabled to improve performance.

Example: AttachToGuard=0

5.1.20 Dialer Detection

The engine can detect dialers and report them like a virus. By default, this option is disabled. To enable dialer detection, set *ReportDialer* to 1.

Example: ReportDialer=1

5.1.21 Macro Virus Heuristics

The engine can detect macro viruses. When this option is enabled, you can also specify how infected macros are treated.

Example: OLEHeuristicEnabled=1
 RemoveSuspiciousMacros=0

 0 = Delete all suspicious or infected macros
 1 = Delete all macros if one macro is suspicious or infected

5.1.22 Win32 File Heuristic

For improved virus detection, you can enable Win32 file heuristic. By default, this option is disabled. If enabled, you can additionally set the sensitivity with a further parameter.

Example: Win32HeuristicEnabled=1
 Win32HeuristicScanMode=0

 0 = Low sensitivity
 1 = Medium sensitivity
 2 = High sensitivity

6 Appendix B – Structure of SAVAPIDL.INI

6.1 Possible Entries in SAVAPIDL.INI

SAVAPIDL.INI is the initialization file for communications between SAVAPI.DLL and the engine.

By default, this initialization file does not exist and the default values are used. To change the default port for communications with the engine, a file named SAVAPIDL.INI must be created in the directory containing the SAVAPI.DLL file. This file is created by terminating the SAVAPI service and contains only the following entry:

```
[SAVAPI2DLL]
PortNumber=18370
```

6.1.1 Port Number

This is the number of the TCP/IP port used for communications between the SAVAPI service and the SAVAPI.DLL. If this port is already assigned, you can change this value. Note, in this case, that you must then also change the corresponding entry in the engine's initialization file (SAVAPI.INI).

Example: PortNumber=18370

7 About GROUP Technologies AG

GROUP Technologies AG is a world leader in E-mail Lifecycle Management. The company's fully integrated iQ.Suite products ensure efficient security and effective organization of e-mail, from encryption, virus protection, and spam filters to e-mail classification and secure archiving.

The iQ.Suite is modular, fully scalable, and offers a high degree of investment security. The modules are completely server-based, can be centrally administered at a low cost, and are available for Lotus Domino, Microsoft Exchange and SMTP platforms.

With the iQ.Suite, companies can reduce costs, optimize the performance of their e-mail environment, and increase productivity. GROUP's clients include many well-known companies such as Deutsche Bank, Ernst & Young, Honda, Heineken, and Miele. More than six million users and 2,000 companies worldwide protect and organize their systems with GROUP Technologies products.

GROUP Technologies AG is headquartered in Karlsruhe. It maintains a subsidiary in the USA, and distributes its products internationally, both directly and through partner companies.

www.group-technologies.com

© 2005 GROUP Technologies AG

The product descriptions are general and descriptive in nature. They can be interpreted neither as a promise of specific properties nor as a declaration of guarantee or warranty. The specifications and design of our products can be changed at any times without prior notice, especially to keep pace with technical developments.

The information contained in this documentation deals with issues as assessed by GROUP Technologies AG at the time of publication. As GROUP Technologies AG is bound to react to changing market requirements, this document by no means represents an obligation by GROUP Technologies AG and GROUP cannot guarantee the correctness of the information presented in this document after its publication.

This documentation is intended for information purposes only. GROUP Technologies AG hereby excludes any warranty, express or implied, for this document. GROUP Technologies AG is unable to guarantee, either explicitly or tacitly, the quality, execution, standardization or suitability for a specific purpose.

All product or company names in this document may be protected brand names of their respective owners.

Headquarters

GROUP Technologies AG

Ottostrasse 4

76227 Karlsruhe / Germany

Phone +49(0)721-4901-0

Fax +49(0)721-4901-199

info.de@group-technologies.com

www.group-technologies.com



North American Headquarters

GROUP Technologies

120 Quarry Drive, Suite B214

Milford, MA 01754/USA

Phone +1 508-473-3332

Phone 877-476-8755 (US and Canada)

Fax +1 508-473-9940

info.us@group-technologies.com

www.group-technologies.com