



AppGate Client for Windows Mobile

Version 1.4.0

AppGate Client for Windows Mobile

Copyright © 2010 AppGate Network Security AB

Table of Contents

1. Getting started	1
1.1. Requirements	1
1.2. Installation	1
1.3. Uninstallation	1
1.4. Launching	1
1.5. User interface	1
2. Basic use	3
2.1. Settings	3
2.2. Commands	3
3. Advanced usage	5
3.1. Roaming	5
3.2. Advanced settings	5
3.3. List of tunnels	6
3.4. List of web accesses	7
3.5. Upload log	7
4. Administration	8
4.1. Supported authentication methods	8
4.2. Supported attributes	8
4.3. Supported client check options	8
4.4. Supported components	10
4.5. Supported connection settings	11
4.6. Supported clustering settings	11
4.7. Launching web pages	11
4.8. Creating browser bookmarks	11
4.9. Integration with remote third-party PKI management	12
4.10. Creating mail accounts	12
4.11. Creating ActiveSync accounts	13
4.12. Launching the Remote Desktop client	15
4.13. Advice on creating services	15
4.14. Configuration file	17
A. Specific customizations	21
A.1. Local customization of client dialogues	21
B. Copyright notices	22
B.1. PuTTY	22

Chapter 1. Getting started

1.1. Requirements

The client can be installed on any device which is based on Windows Mobile 2003 or higher. This includes devices from HTC, Samsung, HP, Palm, Psion, Motorola and Sprint.

The AppGate server being connected to must be of version 6.1 or higher.

This manual requires that you are familiar with the basic concepts of the AppGate system.

1.2. Installation

The recommended method of installation is to use SMS provisioning on the AppGate server. A download link and personal settings will be sent to the device as SMS messages. If provisioning is not an option, the client can also be installed manually.

The recommended method of manual installation is to connect the device to a PC and run the appropriate installer on the PC. The installer to use is `AppGate_PPC2003_140.exe` for Pocket PC 2003 devices and `AppGate_WM_140.exe` for all other devices. An alternative way of installation is to open the appropriate CAB file on the device. The CAB file to use is `AppGate_PPC2003_140.cab` for Pocket PC 2003 devices and `AppGate_WM_140.cab` for all other devices.

For SMS interception to work properly, it is recommended that the device is restarted after installation. The installation file will offer a choice to do so.

Some possible ways to open the file on the device are:

- Over the web, by publishing the file on a web server and opening it from the web browser on the device.
- By copying the file from a PC to the device using ActiveSync, and opening it with the file explorer application on the device.
- On a memory card, from where the file is opened with the file explorer application on the device.
- By sending the file over Bluetooth.

1.3. Uninstallation

To uninstall the client, open the Start menu, then Settings, System, and Remove programs. Select AppGate Client and then Remove.

1.4. Launching

The client is launched by selecting its icon from Programs under the Start menu.

Note that the client is running until **Exit** is chosen in the client or the device is shut down.

1.5. User interface

When the client is running in the foreground, it will display its current status (Connected, Suspended etc.) under the row `Status`, and when a connection has been started, the login time under the row `Login time` and the amount of data transferred in the session under the row `Transferred`.

On the bottom of the display is a menu bar with two choices: **Menu** and **Exit**. **Menu** opens a menu where you can change the login settings, and connect or disconnect the client. **Exit** makes the client disconnect from the server before it exits.

Some of the settings described in this manual may be unavailable in the client GUI if they have been locked down in the configuraion.

Chapter 2. Basic use

2.1. Settings

The client will keep its settings in a file called `settings.txt` (see Section 4.14, “ Configuration file ”). If it is desired to have the client pre configured with certain settings this file may be installed on the device at the same time as the client is installed.

Otherwise, if the `settings.txt` file is not present, or does not contain enough information to make a connection, when the client starts it will look for a settings SMS in the Inbox folder. This mechanism is used by the AppGate Mobile Provisioning feature. If the client finds an SMS the `Status` row will say `Downloading settings . . .` until the settings are downloaded from a server. It is also possible to enter the settings later, or change them later, by opening the settings screen with **Settings...** in the menu.

Settings that are not applicable to the current configuration will not be visible in the menu.

These settings are required to establish a session:

Server address

The host name or IP address of the AppGate server. If the port number is not 22, add a colon and the port number, as in `appgate.example.org:443`.

Login name

Your login name (not real name) for the AppGate server.

Authentication

The client supports the authentication methods **Password**, **Radius**, **SecurID CryptoCard**, **Public key**, and **Certificate**. Note that the AppGate administrator can rename all methods, and there can be more than one Radius method.

Certificate and Private key

When choosing **Certificate** or **Public key** as authentication method, a new setting called **Certificate** or **Private key** will become visible. This setting will have the text **Not installed** or **Installed**, while offering the choice **Install...** or **Uninstall...**, depending on whether the corresponding certificate or private key has been installed.

When choosing **Install...**, the client will ask for a PKCS #12 file (for Certificate) or a PuTTY private key file (for Public key) to authenticate with. It is not possible to authenticate without a valid file.

The certificate or private key file will be copied to a hidden location.

When choosing **Uninstall...**, the installed certificate or private key will be deleted.

2.2. Commands

The following commands can be found under **Menu**:

Connect

Connect to the AppGate server. When the `Status` row says `Connected`, all encrypted tunnels are established and ready for use.

Suspend

Disconnect temporarily from the AppGate server. Does not close the established tunnels, only pause them. The `Status` row will say `Suspended`.

Resume

Reconnect to the AppGate server, if the connection is temporarily down. Traffic in the established tunnels will be resumed. This will not require authentication. The `Status` row will say `Connected` after a successful reconnection.

Stay suspended

Tells the client to stop trying to reconnect to the AppGate server.

Change password

Displays a dialog for changing the login password. This is only available when the client is connected to the AppGate server, and the server allows the ability to change the password. In practise this may happen only if the account is an AppGate Local account or an account stored in Active Directory (with Support for Active directory account information).

It is not possible to change password when using certificate-based or public key-based authentication.

Chapter 3. Advanced usage

3.1. Roaming

Roaming is a way to disconnect and reconnect to the AppGate server without losing data or having to authenticate again. It is also a way to restore a broken connection, and to enhance battery life, as mobile devices tend to use more power when connected to a data network.

A paused or broken connection which can be resumed is called a suspended connection. The connection can be suspended and resumed manually by choosing **Suspend** or **Resume** from the menu, when available.

Roaming is available only if the AppGate administrator has granted you the **Roaming** capability, see the AppGate Security Server manual.

3.2. Advanced settings

The following settings are available in the settings screen which is opened with **Settings...** in the menu:

Network

Can be either **None**, **Automatic** or **User selected**, the default is **Automatic**.

If the choice **None** is selected, it is the user's responsibility to activate the appropriate network. The choice **Automatic** lets the operating system decide the best network to use. This is typically Internet or Work, depending on the server address. When choosing **User selected**, a network has to be selected once, the client will then automatically use this network.

Roaming mode

Can be either **Manual** or **On demand**. Has no effect if roaming is not available.

The choice **Manual** means that the client tries to stay connected, unless the user has chosen to suspend. If the connection breaks, the client will try to resume, and retry every two minutes until it succeeds or **Stay suspended** or **Cancel resume** is chosen from the menu.

The choice **On demand** makes the client suspend the connection after three minutes of inactivity in the encrypted tunnels. It also makes the client resume automatically upon traffic in the encrypted tunnels, typically a request from a web browser or mail client. If the client fails to resume, it will retry every 30 seconds until it succeeds or **Manual** is chosen.

The default setting is **Manual**. This is mainly because IMAP IDLE (also known as "push e-mail") cannot be used with **On demand**, since with **On demand** the client may suspend after three minutes even if there is an IMAP tunnel waiting for traffic from the server.

Keep-alive

Makes the client send "keep-alive" data after every 30 seconds of inactivity, which tends to make the connection more stable. Should only be used when absolutely necessary as it drains the battery and the additional network traffic might be expensive. This is disabled by default.

Compression

Makes the client compress all traffic to and from the AppGate server, which tends to speed up otherwise slow connections. Enabled by default.

Log level

Determines how much information is saved in the log file, which is called `log.txt`.

The log file can be found in the client's installation directory, which typically is `\Program Files\AppGate Client`.

The following log levels are available:

None

No log file is created. This is the default setting.

Brief

Meant for technical users to aid self-diagnosis.

Verbose

Includes large amounts of internal messages. The client runs slower at this log level.

Verbose + Data

Includes all traffic sent in the session (except for AppGate passwords). Be careful when storing or sending such log files! The client runs slower at this log level.

The log file is cleared each time the client is started, so that only the last session is visible. The log file from the previous session is copied to the file `previous_log.txt` in the same directory. When the log file reaches the size of 1 MB, it is overwritten from the beginning, so that only the last 1 MB are stored.

Accept new keys

Makes the client automatically accept the AppGate server's host key when connecting to the server for the first time. This is enabled by default.

3.3. List of tunnels

When the client is connected to a server, the choice **List tunnels** appears under **Advanced** in the menu. It opens a text window which lists all tunnels together with statistics and current status for each tunnel. There is one paragraph of text for each tunnel, which, for instance, can look like this:

```
SMTP Access
From 127.0.0.1 port 25
To mail.company.com port 25
Open 1 Total 4 Rx 2320 Tx 30490
```

Where the fields have these meanings:

First row

Description of the tunnel, set by the administrator.

Second row

Listening address and port on the mobile device. The text `(changed)` indicates that the default port is busy and another port is used instead. The text `Not listening` indicates that the tunnel could not be opened or has been closed for some reason.

Third row

Destination address and port behind the AppGate server.

Open

The number of active instances of this tunnel.

Total

The number of times this tunnel has been opened, including the active instances.

Rx

The number of bytes received through this tunnel.

Tx

The number of bytes sent through this tunnel.

3.4. List of web accesses

When the client is connected to a server, the choice **List web accesses** appears under **Advanced** in the menu. It opens a text window which lists all web sites that are made available through the built-in web proxy on the AppGate server. There are two rows of text per web site, where the first row is a description of the site, and the second is the base address (URL) to the site.

3.5. Upload log

When the client is connected to a server, the choice **Upload log** appears under **Advanced** in the menu. It will send a number of recent log messages to the server. The status row will say `Uploading log...` while the messages are being sent, and `Connected` when finished. This functionality is useful for diagnosing problems together with an administrator. Note that events from previous sessions are not included. The **Log level** setting does not affect which messages are sent or how many.

Chapter 4. Administration

4.1. Supported authentication methods

Password

Support for changing the password.

Radius

No support for changing the password.

One-time passwords delivered over SMS with services such as MidEye or NordicEdge can be caught by the client while displaying a password dialog. The SMS message must be formatted so that there is a colon (:) before the password, then optional whitespace, and then the password, which may not contain whitespace. If any text or character comes after the password, there must be whitespace inbetween. There may not be any other colon in the SMS message. The Radius prompt for the one-time password must contain the text `one time`, `one-time` or `otp` (case-insensitive).

SecurID

Support for changing the password when required.

CryptoCard

Fully supported.

Certificate

Supported for PKCS #12 files, with or without passphrase.

Public key

Supported for PuTTY private key files, with or without passphrase.

4.2. Supported attributes

The client sets the following attributes in the AppGate session:

Table 4.1.

Attribute	Value
<code>ag_client_type</code>	<code>mobile</code>
<code>ag_client_is_webstart</code>	<code>false</code>
<code>ag_client_version</code>	<code>1.4.0</code>
<code>identd</code>	<code>false</code>
<code>persfw</code>	<code>false</code>
<code>iptunneling</code>	<code>false</code>
<code>platform</code>	<code>windowsce.windowsmobile.mobile</code>

4.3. Supported client check options

The client has built-in support for a set of client check options, which are modeled after those in `CHECK.EXE` for the Windows platform. When defining a client check, any binary file can be specified, as it is not used. The supported options are:

-fileexists PATH

True if the regular file *PATH* exists.

-filesize PATH

Returns the size of the regular file *PATH*.

-filenewer PATH DATE

True if the regular file *PATH* exists and has been modified since *DATE*.

-fileolder PATH DATE

True if the regular file *PATH* exists and has not been modified since *DATE*.

-fileageless PATH AGE UNIT

True if the regular file *PATH* exists and its age is less than *AGE UNIT*.

-fileagemore PATH AGE UNIT

True if the regular file *PATH* exists and its age is more than *AGE UNIT*.

-filemd5 PATH

Returns the MD5 sum of the regular file *PATH*.

-direxists PATH

True if the directory *PATH* exists.

-process FILENAME

True if a process called *FILENAME* (including *.exe*) is running.

-imei

Returns the IMEI number of the device.

Windows Mobile devices may add trailing digits after the usual 15 digits.

-imsi

Returns the IMSI number of the SIM card, if any.

-regexists ROOT KEY [VALUE]

True if the specified registry key or value exists.

-regprint ROOT KEY [VALUE]

Returns the contents of the specified registry value or the default value of the specified registry key. Only supported for string and number (DWORD) values.

-filterversion

Returns the version of the installed AppGate Filter, if any.

-badcertcount

Returns an integer count of the number of wrong certificate passphrases entered since the client was installed or the counter was reset.

-privatekeyprotection

Returns a word which tells how the client protects, or would protect, certificate and private key files. The following words are defined:

none

The client provides no protection, the file is visible and readable by anyone who uses the device.

hidden

The file is marked as hidden and is therefore normally not seen by someone who browses the files on the device.

symbian

The file is protected by the Symbian operating system's security measures. For a user to read the file, it would be necessary to either disassemble the device and use advanced electronic

tools to read the internal memory (which may also be encrypted), or to install a special application signed with the closely guarded *AllFiles* capability which only the device manufacturer can grant, or to otherwise compromise the operating system's security measures.

Checks which are either true or false return either *yes* or *no*, so normally one would create an access rule which checks if the check attribute matches "*yes*".

If the client fails to execute a client check, the attribute is not set, but a short error message is written to the attribute `ATTRIBUTE_failure_reason`, where `ATTRIBUTE` is the name of the ordinary attribute.

DATE is given as *YYYYMMDDhhmmss* or any shorter length thereof, for instance *20061231235959* for 23:59:59 the 31st Dec 2006, or *200612* for Dec 2006.

In *AGE UNIT*, *AGE* is a number and *UNIT* is either *days*, *day*, *hours*, *hour*, *minutes*, *minute*, *seconds* or *second*. For instance, *-filenewer /gizmo.dat 12 hours* is true if the file `\gizmo.dat` has been modified or created within the last 12 hours.

Paths should be written with slashes instead of backslashes, as in `/Windows/gizmo.dat`, which refers to `\Windows\gizmo.dat`.

Valid registry roots are `HKEY_LOCAL_MACHINE`, `HKEY_CLASSES_ROOT`, `HKEY_CURRENT_USER`, `HKEY_USERS`, and the abbreviations `HKLM`, `HKCR`, `HKCU`, and `HKU`.

Registry keys should be written with slashes instead of backslashes, as in `Software/Gizmo/Gadget`, which refers to `Software\Gizmo\Gadget`.

4.4. Supported components

IP Access

Supported for TCP on a single port on a single host.

In a client application on the device, the host name of the application server can be used when accessing the tunnel. The addresses *127.0.0.1* and *localhost* can also be used.

Web Access

Supported for one or more ports on one or more hosts per web access component. IP addresses, IP address ranges and port ranges are not supported. Start URL is not supported, but web sites can be launched with client commands.

Note that on Symbian devices it is necessary to configure the web browser to use the built-in web proxy, which is an HTTP proxy at *127.0.0.1* port 80.

Client Command

It is possible to define commands in the format

```
\PATH\APPLICATION.exe ARGUMENTS...
```

to launch arbitrary applications on the device.

The client also recognizes these special commands:

browser

See Section 4.7, "Launching web pages"

bookmark

See Section 4.8, "Creating browser bookmarks"

mailaccount

See Section 4.10, "Creating mail accounts"

syncaccount

See Section 4.11, “Creating ActiveSync accounts”

remotedesktop

See Section 4.12, “Launching the Remote Desktop client”

resetbadcertcount

Resets the counter which is read with the **-badcertcount** client check option.

User Message

Is shown in plain text, where all HTML code is stripped away.

Roaming capability

Fully supported.

All available components are started when logging in, regardless of **Auto start** settings in the containing services.

4.5. Supported connection settings

Some of the server's connection settings require support by the client:

Listen ports

Nonstandard ports fully supported.

Message of the Day

Fully supported.

The setting '**You are already logged in**' warning is not supported.

4.6. Supported clustering settings

Load balancing and failover are fully supported.

4.7. Launching web pages

A web page can be launched in the native browser of the device by defining a client command with the format:

`browser URL`

Example 4.1. Opening localhost port 80 in a browser

Define this client command:

```
browser http://127.0.0.1/
```

4.8. Creating browser bookmarks

A bookmark can be created in the native browser of the device by defining a client command with the format:

`bookmark URL TITLE`

Example 4.2. Intranet for company.com at port 80

Define this client command:

```
bookmark http://127.0.0.1/ "Intranet for company.com"
```

4.9. Integration with remote third-party PKI management

Two client commands can be used to distribute and manage certificates with minimal hassle. This requires a configured third-party PKI server. Both commands take the URL of the server as argument.

```
installcert URL  
requestnewcert URL
```

Example 4.3. Remote initial install of a certificate

Define this client command:

```
installcert http://company.com/pki-handler
```

Add the command to a service with appropriate access rules and all the user needs to do is to log in using a password, the certificate will be automatically installed.

Note that when this command is completed, the client will change authentication method to certificate and exit.

Example 4.4. Remote update of a certificate

Define this client command:

```
requestnewcert http://company.com/pki-handler
```

This command is useful to replace the current certificate with a new one. Add the command to a service with appropriate access rules and the new certificate will be installed when the user logs in.

4.10. Creating mail accounts

A mail account can be created and launched on the device by defining a client command with the format:

```
mailaccount OPTIONS
```

More than one mail account can be created in a single role. A mail account that has already been created with the same parameters is not changed. The options that can be given are:

Table 4.2.

Option	Description
-pop	Use the POP protocol instead of IMAP for incoming mail.
-accountname TEXT	User-friendly account name. If not given, the e-mail address is used.
-emailaddress TEXT	The e-mail address to be displayed in sent messages. If not given, the e-mail address is constructed from the login name and domain name.

Option	Description
<code>-domain TEXT</code>	The domain part of the e-mail address to be displayed in sent messages. If not given, the part of the AppGate login name after "@" is used. If this is not present, the AppGate server address is used, discarding the part until the first dot.
<code>-alias TEXT</code>	The e-mail alias (full name) to be displayed in sent messages. If not given, no alias is used.
<code>-loginname TEXT</code>	The login name for authenticating to the mail servers. If not given, the AppGate login name is used. If the authentication method is Certificate, the client will not know the AppGate login name, and it is necessary to include <code>-loginname %U</code> in the command. The variable <code>%U</code> will expand to the forward account name.
<code>-password TEXT</code>	The password for authenticating to the mail servers. If not given, the user is asked to enter the password when creating the account.
<code>-samepassword</code>	Use the AppGate password for the mail servers. If not given, the user is asked to enter the password when creating the account.
<code>-smtpauth</code>	Enable SMTP authentication when sending mail.
<code>-launch</code>	Launch the mail account in the native mail client. Note that the mail client may or may not synchronize with the server when this happens.

Arguments to options can be enclosed in quotation marks.

Example 4.5. Simplest case

The AppGate server address is `appgate.company.com`, the e-mail addresses have the format `LOGINNAME@company.com`, the e-mail server is an IMAP server, the tunnels listen on the default ports and the mail client should not be launched. Define this client command:

```
mailaccount
```

Example 4.6. Need to define domain, launch mail client

The AppGate server address is `appgate.helsinki.company.com`, the e-mail addresses have the format `LOGINNAME@company.com`, the e-mail server is an IMAP server, the tunnels listen on the default ports and the mail client should be launched. Define this client command:

```
mailaccount -domain company.com -launch
```

4.11. Creating ActiveSync accounts

An ActiveSync account can be created and launched on the device by defining a client command with the format:

```
syncaccount OPTIONS
```

A sync account that has already been created with the same parameters is not changed. The options that can be given are:

Table 4.3.

Option	Description
<code>-accountname TEXT</code>	User-friendly account name. If not given, the e-mail address is used.
<code>-emailaddress TEXT</code>	The e-mail address to be displayed in sent messages. If not given, the e-mail address is constructed from the login name and domain name.
<code>-domain TEXT</code>	The user's domain. If not given, the part of the AppGate login name after "@" is used. If this is not present, the AppGate server address is used, discarding the part until the first dot.
<code>-alias TEXT</code>	The e-mail alias (full name) to be displayed in sent messages. If not given, no alias is used.
<code>-loginname TEXT</code>	The login name for authenticating to the server. If not given, the AppGate login name is used. If the authentication method is Certificate, the client will not know the AppGate login name, and it is necessary to include <code>-loginname %U</code> in the command. The variable <code>%U</code> will expand to the forward account name.
<code>-password TEXT</code>	The password for authenticating to the server. If not given, the user is asked to enter the password when creating the account.
<code>-samepassword</code>	Use the AppGate password for the ActiveSync server. If not given, the user is asked to enter the password when creating the account.
<code>-server TEXT</code>	An IP address or host name where the server can be reached. This must be given if the server is reached through a web access. If not given, <code>localhost.ag</code> will be used.
<code>-ssl</code>	Use SSL when talking to the server.
<code>-nomail</code>	Do not use the account for mail synchronization.
<code>-nocalendar</code>	Do not use the account for calendar synchronization.
<code>-nocontacts</code>	Do not use the account for contacts synchronization.
<code>-launch</code>	Start up the ActiveSync client. Note that it may or may not synchronize with the server when this happens.

Arguments to options can be enclosed in quotation marks.

Example 4.7. Simplest case

The AppGate server address is `appgate.company.com`, the domain is `company.com`, mail, calendar and contacts should be synchronized and the ActiveSync client should not be launched. Define this client command:

```
syncaccount
```

Example 4.8. Need to define domain, launch ActiveSync client

The AppGate server address is `appgate.helsinki.company.com`, the domain is `company.com`, mail, calendar and contacts should be synchronized and the ActiveSync client should be launched. Define this client command:

```
syncaccount -domain company.com -launch
```

Example 4.9. Synchronize only calendar and contacts

Only calendar and contacts should be synchronized and the mail client should not be launched. Define this client command:

```
syncaccount -nomail
```

4.12. Launching the Remote Desktop client

On Pocket PC devices, the built-in Remote Desktop client, also called Terminal Services Client, can be launched on the device by defining a client command with the format:

```
remotedesktop OPTIONS
```

The options that can be given are:

Table 4.4.

Option	Description
<code>-server</code> TEXT	An IP address or host name where the server can be reached. If not given, <code>127.0.0.1</code> will be used.
<code>-port</code> NUMBER	Port number where the server can be reached. If not given, the Remote Desktop client will use the default port.
<code>-width</code> NUMBER	Width of desktop, in pixels. If not given, the Remote Desktop client will use a default value.
<code>-height</code> NUMBER	Height of desktop, in pixels. If not given, the Remote Desktop client will use a default value.

4.13. Advice on creating services

In most cases it is enough to define an IP access component for each server port, where the local port is the same as the remote port. The only special configuration that must be done in third-party applications (if client commands are not used for this purpose) is to set the server address to `127.0.0.1` (except in the ActiveSync client on Windows Mobile 5, where the server address must be `localhost.ag` or another hostname). This is true especially for the mail protocols SMTP (port 25), POP (port 110) and IMAP (port 143), as these ports usually are available on mobile devices. On Windows Mobile, if the remote server address is a hostname (which includes dots so that it belongs to the "Internet" network), such as `imap.company.com`, it will be mapped to `127.0.0.1` in the phone, so `imap.company.com` can be set as server address in the mail client, and the benefit is that the users can configure their applications exactly like they would on a PC.

However, port 80 is used for many purposes, and these conflicts often require creativity from the administrator. In the case of several intranet sites or web applications, it is often enough to assign different local ports to them, such as 8000, 8001 and 8002. The browser would then be pointed to `http://127.0.0.1:8001` or a similar address to access one of these sites.

Some ActiveSync clients, such as the one included in Windows Mobile, cannot be configured to use any other server port than 80 (or 443 if SSL is used, which unnecessarily degrades performance when used together with the AppGate system). It would then be advisable to assign port 80 to the ActiveSync component and other ports to web applications.

Another difficulty is that some web applications (such as Outlook Web Access) require that a fixed hostname, such as `intranet.company.com` is used in the URL, rather than an alias such as `127.0.0.1:8001`. In this case, web access components must be used instead. All web applications, including ActiveSync, will share port 80 and the AppGate server's built-in web proxy will route traffic

to the right destination based on the hostname being used in the URL. There are both advantages and disadvantages with using web access components. On the positive side, Single Sign On (no need to authenticate a second time) is possible, and the AppGate server can conveniently both control and log what the user does. On the negative side, browsers on Symbian phones have to be configured manually to use the built-in web proxy (see "Web Access" under Section 4.4, "Supported components"), and will not be able to surf the internet without a connection to the AppGate server (which, again, may be positive if there is a surfing policy).

Example 4.10. Mail only

There is an IMAP server at *imap.company.com* and an SMTP server at *smtp.company.com*. Create a service with these components:

- An IP access component for TCP from port 25 to port 25 at *smtp.company.com*.
- An IP access component for TCP from port 143 to port 143 at *imap.company.com*.
- A client command which defines and optionally opens the mail account (see Section 4.10, "Creating mail accounts").

Example 4.11. Mail and two internal web sites

There is an IMAP server at *imap.company.com*, an SMTP server at *smtp.company.com* and two internal web sites at *intranet.company.com* and *crm.company.com*. Create a mail service with these components:

- An IP access component for TCP from port 25 to port 25 at *smtp.company.com*.
- An IP access component for TCP from port 143 to port 143 at *imap.company.com*.
- A client command which defines and optionally opens the mail account (see Section 4.10, "Creating mail accounts").

Create a web site service with these components:

- An IP access component for TCP from port 80 to port 80 at *intranet.company.com*.
- Optionally, a client command which opens the web site (see Section 4.7, "Launching web pages"):

```
browser http://127.0.0.1
```

Create another web site service with these components:

- An IP access component for TCP from port 8000 to port 80 at *crm.company.com*.
- Optionally, a client command which opens the second web site (see Section 4.7, "Launching web pages"):

```
browser http://127.0.0.1:8000
```

Example 4.12. Exchange server and one web site

There is an Exchange server at *exchange.company.com* and an internal web site at *intranet.company.com*. Create an ActiveSync service with these components:

- An IP access component for TCP from port 80 to port 80 at *exchange.company.com*.
- A client command which defines and optionally opens the ActiveSync account (see Section 4.11, “Creating ActiveSync accounts”).

Create a web site service with these components:

- An IP access component for TCP from port 8000 to port 80 at *intranet.company.com*.
- Optionally, a client command which opens the second web site (see Section 4.7, “Launching web pages”):

```
browser http://127.0.0.1:8000
```

Example 4.13. Exchange server and one picky web site

There is an Exchange server at *exchange.company.com* and an internal web site at *intranet.company.com*, which does not tolerate being called *127.0.0.1:8000*. Create an ActiveSync service with these components:

- A web access component for port 80 at *exchange.company.com*.
- A client command which defines and optionally opens the ActiveSync account (see Section 4.11, “Creating ActiveSync accounts”). The *-server* option must be set to *exchange.company.com*.

Create a web site service with these components:

- A web access component for port 80 at *intranet.company.com*.
- Optionally, a client command which opens the web site (see Section 4.7, “Launching web pages”):

```
browser http://intranet.company.com
```

4.14. Configuration file

The configuration is stored in a file called *settings.txt*. This file consists of a number of lines of the form:

```
parameter=value
```

The *parameter* keyword is encoded in 7 bit ASCII. The *value* can be encoded in 7 bit ASCII or UTF-8. The file must not contain a byte order mark. If the configuration file begins with a byte order mark it will be interpreted as part of the first parameter keyword which causes that parameter to be silently ignored. Be careful to not edit the configuration file with an editor that adds a byte order mark to UTF-8 files.

When the client starts it will load the configuration as follows:

- The configuration from the previous session is loaded, if any.

- If the setting `ag_sms_provision` is set to `true` (the default) and if the configuration does not yet have enough information for the client to connect to a server then the client looks for a configuration SMS. See Section 2.1, “Settings”.

The configuration file can contain the following parameters:

Table 4.5.

Parameter	Values	Description
<code>ag_accept_keys</code>	<code>true</code> <code>false</code>	See Accept new keys . Default: <code>true</code> .
<code>ag_accept_keys_lock</code>	<code>true</code> <code>false</code>	If <code>true</code> the user can not change the Accept new keys setting and can not accept new host keys via the client GUI. Default: <code>false</code>
<code>ag_access_point</code>	<code>manual</code> <code>automatic</code> <code>userdefined</code>	See Network . Default: <code>automatic</code>
<code>ag_access_point_id</code>	<code>string</code>	When <code>ag_access_point</code> is <code>userdefined</code> , this parameter specifies which access point to use. See Network . No default.
<code>ag_access_point_lock</code>	<code>true</code> <code>false</code>	If <code>true</code> the user can not edit the Network setting. Default: <code>false</code>
<code>ag_ask_username</code>	<code>true</code> <code>false</code>	If <code>true</code> the user will always be asked for a username during password authentication. Default: <code>false</code>
<code>ag_auth_method</code>	<code>password</code> <code>securid</code> <code>publickey</code> <code>certificate</code> <code>radius</code> <code>cryptocard</code>	See Authentication . Default: <code>password</code>
<code>ag_auth_method_lock</code>	<code>true</code> <code>false</code>	If <code>true</code> the user can not change the Authentication setting. Default: <code>false</code>
<code>ag_auth_method_names</code>	<code>method:name,...</code>	A comma separated list of mappings of the authentication methods to names. The name is what is displayed to the user when choosing authentication method. Default: <code>password:Password,radius:Radius,se-</code>

Parameter	Values	Description
		<i>curid: SecurID, cryptocard: Cryptocard, publickey: Public key, certificate: Certificate</i>
ag_bad_cert_count	<i>integer</i>	The number of wrong certificate passphrases entered since the client was installed or the counter was reset. See -badcertcount . Default: <i>0</i>
ag_bad_cert_count_limit	<i>integer</i>	If the number of failed attempts to unlock the certificate with an invalid passphrase (see -badcertcount) reaches this number then the certificate will be deleted from the device. If this parameter is <i>-1</i> then the number of attempts is not limited, otherwise the counter will be reset to <i>0</i> when a good passphrase is entered. Default: <i>-1</i>
ag_certificate_lock	<i>true</i> <i>false</i>	If <i>true</i> the user can not choose a new certificate via the client GUI. Default: <i>false</i>
ag_compression	<i>true</i> <i>false</i>	See Compression . Default: <i>true</i>
ag_compression_lock	<i>true</i> <i>false</i>	If <i>true</i> the user can not change the Compression setting. Default: <i>false</i>
ag_file_provision_certificate	<i>true</i> <i>false</i>	If <i>true</i> the client looks for a certificate named <i>certificate.dat</i> in the AppGate directory in the internal memory on the phone. This is done when the client starts, when connecting to a server, when the user opens settings and when the client exits. Default: <i>false</i>
ag_keep_alive	<i>true</i> <i>false</i>	See Keep-alive . Default: <i>false</i>
ag_keep_alive_lock	<i>true</i> <i>false</i>	If <i>true</i> the user can not change the Keep-alive setting. Default: <i>false</i>
ag_last_role	<i>string</i>	Last chosen role. No default.

Parameter	Values	Description
ag_log_level	none brief verbose verboisedata	See Log level . Default: <i>none</i>
ag_log_level_lock	true false	If <i>true</i> the user can not change the Log level setting. Default: <i>false</i>
ag_login_name	<i>string</i>	See Login name . No default
ag_login_name_lock	true false	If <i>true</i> the user can not change the Login name setting. Default: <i>false</i>
ag_motd_digest	<i>string</i>	Internal variable keeping track of the last message of the day.
ag_numerical_passwords	true false	If <i>true</i> passwords and certificate passphrases will be entered in numerical mode, similar to when entering a PIN code. Default: <i>false</i>
ag_private_key_lock	true false	If <i>true</i> the user can not choose a new private key for public-key authentication via the client GUI. Default: <i>false</i>
ag_roaming_mode	manual automatic	See Roaming mode . Default: <i>manual</i>
ag_roaming_mode_lock	true false	If <i>true</i> the user can not change the Roaming mode setting. Default: <i>false</i>
ag_server_address	<i>string</i>	See Server address . No default
ag_server_address_lock	true false	If <i>true</i> the user can not change the Server address setting. Default: <i>false</i>
ag_sms_provision	true false	If <i>true</i> the client will look for a configuration SMS and try to download settings, if additional settings are needed in order to connect to the server. See Section 2.1, “Settings”. Default: <i>true</i>
hostkey_server:port:type	<i>host key</i>	The host key of type <i>type</i> for <i>server</i> when connecting to port <i>port</i> .

Appendix A. Specific customizations

A.1. Local customization of client dialogues

Via an additional group of entries in the settings file it is possible to customize a small subset of the client dialogue strings, currently that is mainly strings used for certificate management.

Some of them are used on their own, others are combined to create phrases.

The strings can be encoded in 7 bit ASCII or UTF-8 and will be truncated at a length of 140 bytes. UTF-8 uses up to four bytes for some characters, so those strings may get shorter.

Table A.1.

Parameter	Default	Used at
ag_msg_certerr	"Certificate error"	Error message when a login by certificate failed.
ag_msg_toomanyfail	"Too many failures"	Combined to form "Too many failures, certificate deleted" and
ag_msg_certdeleted	", certificate deleted."	"Too many failures. Failed to delete certificate" messages when the certificate should be deleted
ag_msg_certdelfail	". Failed to delete certificate."	due to too many passphrase errors
ag_msg_passphrase	"Passphrase"	Combined to form "Passphrase", "Passphrase (X more tries)" and
ag_msg_tries1	" ("	"Passphrase (last try)" prompts for the certificate passphrase
ag_msg_tries2	" more tries)"	
ag_msg_lasttry	" (last try)"	
ag_msg_cert_install_error	"Certificate not installed."	Certificate enrollment failure message.
ag_msg_cert_install_success	"Certificate successfully installed."	Certificate enrollment success message.
ag_msg_cert_update_prompt	"Request new certificate?"	Prompt to launch certificate renewal.
ag_msg_cert_update_error	"Certificate not updated."	Certificate renewal failure message.
ag_msg_cert_update_success	"Certificate successfully updated."	Certificate renewal success message.
ag_msg_new_passphrase	"Enter new passphrase"	Prompt for the passphrase to be used in the new certificate.
ag_msg_new_passphrase_again	"Enter new passphrase again"	Prompt to reenter the new certificate passphrase.
ag_msg_passphrase_unacceptable	"New passphrase not accepted, try again?"	Prompt to try again when the passphrase to be used in the new certificate was rejected.
ag_msg_username_prompt	"Enter user name"	Username prompt for password authentication.
ag_msg_password_prompt	"Password"	Password prompt for password authentication.

Appendix B. Copyright notices

The AppGate client is based on PuTTY. The license for PuTTY requires the following notice to be bundled with all copies of the client.

B.1. PuTTY

PuTTY is copyright 1997-2004 Simon Tatham.

Portions copyright Robert de Bath, Joris van Rantwijk, Delian Delchev, Andreas Schultz, Jeroen Massar, Wez Furlong, Nicolas Barry, Justin Bradford, Ben Harris, Malcolm Smith, and CORE SDI S.A.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.