Configuring MassTransit for the Web

By Janie Longfellow, Lorrin Nelson
1/15/2004

Group Logic Technical Support

This document describes how to configure the MassTransit Remote Administration and Web Client features under Apache httpd (including OS X web sharing), Microsoft IIS, and 4D WebSTAR. Unencrypted as well as SSL configuration for all web servers is discussed. In addition to having the web server communicate with the web browser via SSL encrypted https, it is also possible to have the MassTransit Assistant communicate with the MassTransit server with SSL encryption. This distinct feature is also covered in this document.

Table of Contents

Overview

MassTransit Remote Administration and Web Client access involves six components. These are the MassTransit server, the web server, the CGI, the web browser, the MassTransit plug-in, and the MassTransit Assistant. The former three components reside on the server; the latter three reside on the user's computer. Generally speaking, the web server and the CGI cooperate to communicate with the web browser and plug-in, and the MassTransit server communicates with the Assistant. These two communication channels

may be independently configured to use SSL encryption or not, though most configurations will use SSL for both or for neither.

Enabling Remote Administration and Web Client access requires configuring the web server to serve the web pages located in `Remote Administration/MassTransit Remote Admin` inside the MassTransit directory and configuring the CGI via the `mtadmin.cfg` configuration file.

This document frequently refers to tasks such as opening command or terminal windows, fixing line endings, etc. Many of these concepts are explained at the end of this document.

SSL Overview

To use SSL, a PEM format x509 certificate is required. A certificate consists of a certificate file and a key file. The certificate is used both to encrypt data being sent and as a form of identification. There are three steps to obtaining a certificate. These three steps are carried out differently for each web server, but the purpose of each step remains the same.

1. Generate a private server key. This key is later used to encrypt the outgoing data.
2. Generate a CSR (certificate signing request). The CSR is linked to the key, the identity of the server's owner, and the URL of the server. **The server URL is stored in the common name (CN) field of the certificate.**
3. Obtain CA (certificate authority) signature. The CA signs the CSR after verifying that the holder of the CSR and private server key matches the identity specified in the CSR. **The signed CSR is the certificate**. Because the receiving party trusts the CA, the CA signature proves to the receiving party that the certificate holder really is the party named in the certificate.

There are three ways to sign the CSR. The first is to have it signed by a publicly known Root CA such as Verisign or Thawte. This is optimal, since these Root CAs are known and trusted.

The second alternative is to have another CA, such as an in-house IT department or a lesser known 3rd party CA, sign the CSR. When using a CA that is not well known, it is necessary to distribute the CA's certificate to clients. Keep in mind that the CA's certificate is independent of the server's certificate! Without the CA's certificate, the receiving party cannot trust the CA and therefore cannot assign any validity to the CA's signature on the server certificate.

CA certificate distribution can be done easily via the web server itself or with MassTransit, but to guarantee security the fingerprint of the certificate must be communicated securely and verified. If the receiving party is able to verify the fingerprint of the CA certificate, then the recipient knows she or he has an authentic CA certificate and not a spoof. The task of distributing CA certificates to web browsers is complicated by the fact that different browsers expect the CA certificate to be distributed in different formats. The default format for keys, requests, and certificates is PEM. Some older versions of Internet Explorer, including IE 5.1 Mac, will only accept CA certificates in DER format. The process of distributing a CA certificate is similar for all web servers; see the "Distributing CA Certificates" section for more information.

The final method of signing the CSR is to self-sign it with the private server key. **A self-signed certificate allows encrypted communication but provides no guarantee whatsoever that the holder of the certificate has any connection to the identity specified in the certificate.** Without proof of identity the client cannot distinguish between communications with the true server and a spoof. As such, self-signed certificates do not offer true security and should only be used for testing purposes. Microsoft IIS cannot use a self-signed certificate.

All web server sections below refer to obtaining a CA signature as a single-step process. For explanations of both how to create CA signatures as well as how to self-sign certificates, see the "OpenSSL" section.

Web Server Certificates vs. MassTransit Server Certificates

Generally speaking it is possible, and desirable, to have the web server and the MassTransit server use the same certificate. However, some web servers, such as WebSTAR and IIS, maintain strict control over the CSR generation process and hide the server's private key. In this situation the MassTransit server has to have a separate certificate.

The MassTransit certificate serves a somewhat different purpose than the web server certificate. Because the web server and the MassTransit server cooperate closely when communicating with web clients, there is no need for the MassTransit Assistant to verify the identity of the MassTransit server: the MassTransit server is automatically known to be the same entity as the web server. For communication with web clients it therefore is inconsequential whether the MassTransit server uses a CA signed certificate or an automatically generated, self-signed one.

In communication between two MassTransit servers, however, the web server is not involved and cannot act as a proof of identity. To have truly secure communication here requires that the MassTransit servers use CA signed certificates.

User Considerations (Mac OS X)

For security reasons, the web server and MassTransit should run as a regular user instead of an administrator or as `root`. However there are some limitations that sometimes prevent this:

- With versions of Mac OS X prior to 10.3 (Panther) it is not possible for the web server and MassTransit to run as different users.
- MassTransit must run as `root` in order to listen on ports below 1024. (The web server does not need to run as `root` to serve data on ports below 1024).
    - Because FTP uses port 21, MassTransit must run as `root` to listen for FTP calls.
- MassTransit runs as the user that is logged in when it is launched.
- Apache runs as `www` by default. Since it is not possible to log in as `www` MassTransit cannot run as this user.

- WebSTAR runs as `webstar` if that user exists on the system. Otherwise it displays the error message "`Unable to get user info!`" and runs as the currently logged in user.

The following table lays out the recommended configurations for each combination of Mac OS X, desired port usage, and web server.

Operating System	MassTransit ports	Web Server	Web Server should run as	MassTransit should run as
10.3 (Panther)	> 1024 only	Apache	`www` or Regular user	Regular user
10.3 (Panther)	> 1024 only	WebSTAR	`webstar`	`webstar`
10.3 (Panther)	Any	Apache	`www`	`root`
10.3 (Panther)	Any	WebSTAR	`webstar`	`root`
pre 10.3	> 1024 only	Apache	Regular user	Regular user
pre-10.3	> 1024 only	WebSTAR	`webstar`	`webstar`
pre-10.3	Any	Apache	This configuration is not possible	
pre-10.3	Any	WebSTAR	`root`	`root`

If you have not already created the user the web server should run as, do so now. Create a regular account using `System Preferences -> Accounts`. Note: the `www` and `root` accounts already exist and do not need to be created!

If MassTransit will not run as `root`, then the user MassTransit runs as should also be the owner of the MassTransit directory hierarchy. If MassTransit is not yet installed, use this user to install MassTransit. If MassTransit is already installed, follow these steps:

1. Open a terminal window
2. Go to the MassTransit directory. **Be sure you are in the correct directory before continuing.**
3. Run the command `chown -R username *` (where username is the user the web server will run as)

Configuring the CGI

The information in this section applies to all web servers. The CGI is the link between the web server and the MassTransit server.

Within the MassTransit folder, open "Remote Administration/MassTransit Remote Admin/mtadmin.cfg" in a text editor. At minimum, configure the following settings:

1. WEB_SERVER_ADDRESS should be set to the DNS name (or IP address) and port clients will use to access the web server. Industry standard is to use port 80 for regular http and port 443 for SSL encrypted https.
2. WEB_SERVER_SECURE must be set to TRUE if SSL is to be used for communication with Web Clients and FALSE if not. To simultaneously serve SSL and non-SSL traffic requires two installations of the MassTransit Remote Admin folder, each with their own mtadmin.cfg and CGI executable, and is currently not officially supported.
3. APACHE_MODE must be set to TRUE for Apache web servers (including Mac OS X Personal Web Sharing) and FALSE for all others.
4. USE_SEND_PARTIAL_EVENTS must be set to TRUE for 4D WebSTAR and FALSE for all others.

Configuring MassTransit

The information in this section applies to all web servers.

In the MassTransit Configuration screen, set the MassTransit to accept incoming TCP/IP calls on a port that is different than the port that will be used by the web server. For most installations, this means accepting TCP/IP calls on port 50000 if SSL is not to be used (with the web server on port 80) or accepting TCP/IP Secure calls on port 444 if SSL is to be used (with the web server on port 443). Note: If the server is configured with two IP addresses ("IP multihoming") it is often preferable to run all traffic over ports 80 or 443, using one IP address for the web server and one address for MassTransit. By default MassTransit listens for traffic on all IP addresses. See the "Configuring MassTransit for Multihoming" Technote for information on how to restrict MassTransit to only listen on certain IP addresses.

SSL is enabled on a per-contact basic. For contacts that require secure communication, bring up the Contact Information dialog and switch to the security tab. Enable "Use Secure Connection To Transfer Files" and choose the desired encryption level.

If the MassTransit server is to use a CA-signed certificate instead of an auto-generated self-signed certificate, select "Use certificate issued by Certificate Authority" and select the private key file and certificate files.

Even if the certificate has been signed by an in-house or lesser known 3rd party CA, it is not necessary to import the CA's certificate; MassTransit implicitly trusts the signature on its own server certificate. However, any remote MassTransit server with which SSL communication is desired must import the CA's certificate. To import a CA certificate click the "Import…" button. Well-known Root CA certificates come pre-installed and do not need to be imported.

- For information on using CA-signed certificates vs. auto-generated self-signed certificates, see the "SSL Overview" section.
- For information on generating certificates for MassTransit, see the "Using the MassTransit CSR Tool" section.

- For information on delivering an in-house CA certificate to a remote MassTransit server, see the "Distributing CA Certificates" section.

Please see the main MassTransit documentation for more information on the MassTransit dialogs and on creating Address Book entries. Address Book entries are where minimum levels of security are specified.

Apache httpd (including Mac OS X Personal Web Sharing)

The information in this section applies only to the Apache web server. Under Mac OS X Personal Web Sharing is implemented with Apache and is covered in this section. On Mac OS X Server 10.2, the setup is identical to Mac OS X. **Note that if you are using Mac OS X Server 10.3, you should instead look at the "Apache httpd for Mac OS X Server" section.**

Before continuing, complete:

- "User Considerations (Mac OS X)"
- "Configuring the CGI"
- "Configuring MassTransit."
- If SSL is to be used, also complete "Using the MassTransit CSR Tool" and obtain a CA signature, as directed by that section.

Because of Apache's great flexibility, there are many ways to configure it. The "Background: httpd.conf layout and organization" subsections provides information on alternate means of configuration and how to decide which method is most appropriate. The other sections provide step-by-step instructions for the most common configurations.

Background: Locating Apache on the Internet (Windows only)

Since Mac OS X comes with Apache pre-installed this section only applies to Windows.

Apache httpd (or simply Apache) is maintained and distributed by the Apache Foundation. See the download links[2] at http://httpd.apache.org/ to obtain the Apache server. The Apache manuals are available at http://httpd.apache.org/docs/ (Apache 1.3) and http://httpd.apache.org/docs-2.0/ (Apache 2.0). These manuals are the definitive reference for configuring Apache.

SSL functionality requires an SSL-capable build of Apache. The Apache Foundation does not distribute Windows binaries with SSL capability. Generally it is possible to find binaries on the web and avoid having to compile the source by hand. As of this writing, SSL-capable Apache 1.3.27 and 2.0.43 binaries are available at http://hunter.campbus.com/. Older versions are available at http://www.modssl.org/contrib/. Normally these binaries include no installer and should be unpacked on top of an existing Apache installation to give it SSL capability. Because there is no installer, the following issues need to be attended to:

[2] The .MSI installers are the most convenient way to obtain an Apache binary. If the .msi is stored with an .msi.exe extension, however, it must be renamed to have a .msi extension before it can be used.

- When unpacking on top of installation that has already been customized, it is necessary to first backup configuration files (primarily httpd.conf) and then to merge them by hand.
- The configuration file ssl.conf is missing from the Apache 1.3.27 binary at hunter.campbus.com. Download the Apache 2.0.43 binary as well and place the ssl.conf from that archive alongside httpd.conf in the Apache 1.3 conf/ directory.
- When Apache is installed with an installer, all the paths in httpd.conf are correctly set to point to the Apache directory. When unpacking an SSL binary, all the paths will be set to a default location, such as f:/apache/. These should all be adjusted to be accurate; not all of them are needed but Apache may not load if it detects invalid paths in its configuration files. This needs to be done in both httpd.conf and ssl.conf.

Downloading and Installing Apache (Windows only)

If the site has changed and these step by step instructions do not apply, consult the "Background: Locating Apache on the Internet" section above.

1. Go to http://httpd.apache.org/download.cgi
2. Choose "Win32 Binary (MSI Installer) from the Apache 2.x section in the middle of the page.
3. Run the installer and fill in the requested server information

Additional steps for SSL installations:

4. Go to http://hunter.campbus.com/ and download the latest Apache 2 SSL enabled binary.
    a. For Apache 1.3.x, download both the Apache 1.3 binary and the OpenSSL binary.
5. Extract all files form the archive (using a utility such as WinZIP), copying on top of the Apache directory created in step 3.
    a. For Apache 1.3.x, extract `openssl.exe`, `libeay32.dll`, and `ssleay32.dll` from the OpenSSL archive and place those in the Apache `bin` directory.
6. Copy `ssleay32.dll` and `libeay32.dll` to the `System32` directory inside the Windows directory.

Installing the Certificate and Private Key

1. Create a directory to store the certificate and key files:
    a. On Mac OS X, as root, create the directories `/etc/httpd/ssl.key/` and `/etc/httpd/ssl.crt`.
    b. On Windows, create directories named `ssl.key` and `ssl.crt` inside the `conf` directory in the newly created Apache folder.

2. Copy the private server key to the `ssl.key` directory created in step 1, changing its name to `server.key`. (This is the file named `mt_private_key.pem` created by the MassTransit CSR tool.)
3. Copy the server certificate returned from the CA to the `ssl.crt` directory created in step 1, changing its name to `server.crt`. (The unsigned CSR generated by the MassTransit CSR tool will not work.)
4. Determine whether to store the private server key in an encrypted or unencrypted state. This decision is critical to maintaining security.

   Generally private server keys are stored in an encrypted form such that a password is required to use them. This is the format the MassTransit CSR tool stores the key in. To avoid having to enter the password each time Apache is started it is possible to store the key in decrypted form, but this should only be done if access to the machine by unauthorized users can be prevented. To be clear, if an unauthorized person obtains access to the unencrypted key, the certificate and key are compromised and must be re-generated and resigned by a CA.

   As of this writing, Apache 2 has no built-in means to allow a passphrase to be entered. Either store the private server key in decrypted form or use an external passphrase entry program Group Logic can assist in the creation of an external program.

   Apache 1.3, which is what Mac OS X Personal Web Sharing uses, does have a built-in mechanism for entering a passphrase. However it requires that Apache always be launched from a terminal window and not by enabling Personal Web Sharing in the System Preferences.

   The relevant configuration setting, `SSLPassphraseDialog`, is discussed in greater detail below.

   If the key will be stored in a decrypted form, decrypt the key now. See the OpenSSL section of this document for instructions on how to decrypt the key.

For more information, please see the mod_ssl homepage at http://www.modssl.org/, the OpenSSL homepage at http://www.openssl.org/, and the Slacksite tutorial on Apache and mod_ssl (albeit geared towards Slackware Linux) at http://slacksite.com/apache/certificate.html.

Background: httpd.conf layout and organization

Apache's main configuration file is `httpd.conf`. On Windows it is located inside the `conf` directory in the newly created Apache folder. On Mac OS X it is located in `/etc/httpd/`. `httpd.conf` has three sections:

1. Global environment settings including loading modules
2. "Main" server settings, which are used directly if no virtual servers are defined or serve as defaults if virtual servers are defined.
3. Virtual host definitions (optional)

Generally any setting that can be specified in the main section can also be specified inside a virtual host definition and vice-versa. Settings specified in a virtual host section are only available to that virtual host, whereas settings specified in the main section are available everywhere. If Apache will be used to serve multiple web sites, virtual hosts are required. If it will only serve MassTransit they are optional. On both Mac OS X and Windows the default non-SSL Apache configuration file has virtual hosts disabled, while the default SSL configuration has them enabled. It is generally easiest to follow this convention.

Additional information on configuration Apache with SSL on Mac OS X is available at http://developer.apple.com/internet/macosx/modssl.html. For Apache with SSL on Windows information as well as current links to binary distributions are available in "The Apache + SSL on Win32 HOWTO". Version 1.6.4 of the HOWTO, which covers Apache 1.3, is available at http://tud.at/programm/apache-ssl-win32-howto.php3.Version 1.6.5 of the HOWTO, which covers Apache 2.0, is available at http://raibledesigns.com/tomcat/ssl-howto.html.

Configuring Apache: httpd.conf - global settings

Use a text editor to open `httpd.conf`, (in `/etc/httpd/` on Mac OS X and inside the `conf` folder in the newly created Apache folder on Windows). On Mac OS X you will need to become root to edit this file and make sure you edit the file with an editor that supports Unix-style line endings. Adjust the following settings within httpd.conf:

ServerRoot (Windows only)

`ServerRoot` needs to be set to point to the directory in which Apache is installed. For example:

```
ServerRoot "C:/Program Files/Apache Group/Apache2"
```

Normally Server Root is correctly set by the installer. If an SSL enabled binary was installed (see "Downloading and Installing Apache"), ServerRoot will be set to a default value, such as `"f:/apache"`. In this case, **do a search and replace throughout all of httpd.conf** replacing the default value with the correct Apache location. Note that **paths must be specified using a Unix-style standard slash '/'**, not a Window-style backslash.

On Mac OS X ServerRoot should already be correctly set to `"/usr"`.

Listen and Port (non-SSL only)

For non-SSL setups, no changes should be needed. On Mac OS X, the following line should already be present:

```
Port 80
```

On Windows, the following line should already be present:

```
Listen 80
```

If it is commented out, uncomment it.

Listen and Port (SSL only)

Determine if Apache should serve unencrypted data in addition to SSL traffic. This is necessary if the server will be used to distribute in-house or lesser known 3rd party CA certificates. See the "Distributing CA Certificates" section for more information.

For SSL setups on Mac OS X, `Port 80` should be commented out (change to `#Port 80`). You will need to add the following just below where the Port directive was:

```
## SSL Support
##
## When we also provide SSL we have to listen to the
## standard HTTP port (see above) and to the HTTPS port
##


<IfModule mod_ssl.c>
   Listen 443
   Listen 80
</IfModule>
```

(Leave out the `Listen 80` line if no unencrypted traffic will be served).

On Windows, comment out the `Listen 80` line if unencrypted traffic will not be served. Note: for Windows **do not add a Listen line for the SSL port here**: it will be specified in the `ssl.conf` file, which is covered later on.

Listen and Port (multihoming setups)

For servers with IP multihoming, configure as above but specify the IP address Apache should use on all Listen lines. For example:

```
Listen 192.170.2.1:80
```

Use multiple lines to allow listening on multiple ports or IP addresses:

```
Listen 192.170.2.1:80
Listen 192.170.2.5:8000
```

Mac OS X users should comment out the existing `Port 80` line and replace it with a `Listen` line in order to specify IP addresses.

mod_ssl (only required for SSL)

Uncomment the line that loads the `mod_ssl` module. On Mac OS X:

```
LoadModule ssl_module libexec/httpd/libssl.so
```

On Windows:

```
LoadModule ssl_module modules/mod_ssl.so
```

For Mac OS X and Apache 1.3 on Windows, the following must be added to the end of the AddModule section as well:

```
AddModule mod_ssl.c
```

User and Group setting (Mac OS X only)

If MassTransit will run as `root`, User and Group can be left as:

```
User www
Group www
```

Otherwise set as below, replacing `username` with the correct user:

```
User username
Group staff
```

See the "User Considerations (Mac OS X)" section for more information.

Background: httpd.conf - main settings

The settings described here may be specified in either in the main section of `httpd.conf` or within `VirtualHost` blocks. These settings are likely to be the same for the whole server and as such are most easily specified in the main section of the httpd.conf.

Configuring Apache: httpd.conf – main settings

ServerAdmin

`ServerAdmin` should be set to the email address of the MassTransit or Web Server administrator. This address is given to users when unexpected errors occur on the server.

ServerName

`ServerName` should be set to the domain name (or IP address if no domain name is available) of the Apache server. In particular, **ServerName should be set to the URL users are going to use in their browsers and must match the Common Name (CN) set in the CSR.** See “Using the MassTransit CSR Tool" above for information on creating SSL certificates.

On Mac OS X **without SSL** this setting can be left commented out.

<Directory>

The default Apache install creates a `<Directory>` block for the default document root directory. It looks like this on Mac OS X:

```
#
# This should be changed to whatever you set DocumentRoot
to.
#
<Directory "/Library/WebServer/Documents">
```

And like this on Windows:

```
#
# This should be changed to whatever you set DocumentRoot
to.
#
<Directory "f:/apache/htdocs">
```

There are many `<Directory>` blocks in `httpd.conf`, but the correct one can be identified by the preceding comment line. Change the path

to point to the MassTransit “Remote Administration” directory. Note that even on Windows paths must be specified using a Unix-style standard slash '/' and not a Window-style backslash. The resulting line should similar to this on Mac OS X:

```
<Directory "/Applications/MassTransit Folder/Remote
Administration/MassTransit Remote Admin">
```

And to this on Windows:

```
<Directory "C:/Program Files/MassTransit/Remote
Administration/MassTransit Remote Admin">
```

Options for <Directory>

The `Options` line in the MassTransit `<Directory>` block must have the following options declared:

```
Options Indexes FollowSymLinks MultiViews ExecCGI
```

DirectoryIndex

`default.html` should be added as a possible index file. Other index files may or may not be listed as well.

```
DirectoryIndex index.html index.htm default.html
```

Note: The DirectoryIndex command may appear within a directory block, causing it to only modify that directory block, or outside causing it to set the default for all directory blocks. It may or may not appear within an IfModule check for mod_dir.c.

Hide *.cfg files

Disable remote viewing of *.cfg files. Change:

```
<Files ~ "^\.ht">
```

to:

```
<Files ~ "^\.ht|^\.cfg">
```

Removing extra Directory blocks

The default Apache httpd.conf includes approximately 50 lines defining other directory blocks and aliases to point to them (/icons/, /manual/, and /cgi-bin/). These have nothing to do with MassTransit and should be removed unless the web server will also be used for another purpose that requires them.

Include conf/ssl.conf (Windows SSL only)

At the bottom of the main section of httpd.conf, the following should appear:

```
<IfModule mod_ssl.c>
    Include conf/ssl.conf
</IfModule>
```

If it is missing, add it by hand. Frequently Apache 2 SSL distributions already include this line while Apache 1.3 SSL distributions do not.

Background: host specific settings

If virtual hosts are being used, the following settings should be specified within the MassTransit virtual host block. Otherwise these settings should be specified in the main section of httpd.conf. **If you are using SSL, proceed to the next section to establish your <VirtualHost> blocks and then return to this section to configure them.**

Configuring Apache: host specific settings

VirtualHost (Virtual host configurations only)

Each virtual host requires a `VirtualHost` block. To set up a virtual host on any platform except Windows SSL, create a `VirtualHost` block at the bottom of httpd.conf. For Windows SSL, **save and close httpd.conf and open** `ssl.conf` and locate the `VirtualHost` block there.

The VirtualHost block should look like this:

```
<VirtualHost *:80>
…
</VirtualHost>
```

Where `*` (or `_default_`) may be replaced by a specific IP address for the host to listen on and `80` should be set to the port to be used.

Document Root

Change (or add, if this is a new virtual host block) the document root to point match the directory specified in the <Directory> block made earlier. Once changed, it should look similar to the line below on Mac OS X:

```
DocumentRoot "/Applications/MassTransit Folder/Remote
Administration/MassTransit Remote Admin"
```

And to the line below on Windows:

```
DocumentRoot "C:/Program Files/MassTransit/Remote
Administration/MassTransit Remote Admin"
```

cgi-script handler

The cgi-script handler must be added as the handler for the CGI files. Add the following either to the existing `AddHandler` lines in the main section or to the virtual host block:

```
AddHandler cgi-script .cgi
```

On Mac OS X, also add:

```
AddHandler cgi-script .acgi
```

This completes the Apache configuration for non-SSL installations.

Configuring Apache: Changes for SSL installations

Note that to configure SSL, you will need a properly generated and signed certificate and key. See the "Generating SSL Certificates" and "Signing SSL Certificates" sections for information on how to obtain a certificate and key.

For Mac OS X, go to http://developer.apple.com/internet/macosx/modssl.html. Towards the bottom is an `<IfModule mod_ssl.c>` block (approximately 60 lines long). Follow the instructions on that page to insert the block at the bottom of `httpd.conf`.

Important note for Mac OS X 10.2.x (Jaguar) users: One of the OpenSSL security patches issues by Apple rendered Apache incapable of serving SSL traffic in a CGI environment. There are three options:

1. Upgrade to Mac OS X 10.3 (Panther).
2. Downgrade the affected file, `libssl.so`, to the old version. (Note that this old version will be missing the security fixes of the new version!). See http://ganter.dyndns.org/misc/apple_ssl.php as well as http://www.macosxhints.com/article.php?story=20030402004719491&query=ssl+apache.
3. Install a fresh copy of Apache instead of using Mac OS X Personal Web Sharing. This document does not explicitly cover that approach, but the information supplied here should be applicable.

For Windows, close `httpd.conf` and open up `ssl.conf`.

The following additional changes need to be made in `ssl.conf` for SSL installations.

<IfDefine SSL> (Windows only)

The entire ssl.conf file is wrapped in an IfDefine block. Ensure that the tag reads `IfDefine SSL`.

SSLPassPhraseDialog

This setting depends on the decision made earlier during "Installing the Certificate and Private Key" regarding whether or not to store the private server key in encrypted form or not.

If an unencrypted private server key is used this setting is irrelevant.

With an encrypted key and Apache 1.3 (this includes Mac OS X) either `SSLPassphraseDialog builtin` or an external program may be used. This offers the security of storing the key in an encrypted form but requires that Apache be started from the command line and the passphrase be entered by hand.

With an encrypted key and Apache 2 an external program must be used.

Use the following syntax to specify an external program:

```
SSLPassphraseDialog exec:/path/to/program/program.exe
```

See http://www.modssl.org/docs/2.8/ssl_reference.html#ToC2 for more information.

SSLSessionCache

SSLSessionCache is required for compatibility with certain versions of Internet Explorer, including all Macintosh versions through at least 5.1. Verify that `SSLSessionCache none` is commented out (or not present) and `SSLSessionCache dbm:logs/ssl_scache` is uncommented.

SSLMutex (Windows only)

Change `SSLMutex` from `file` to `sem`:

```
SSLMutex sem
```

On Mac OS X the line should be left as:

```
SSLMutex file:/var/run/ssl_mutex
```

Certificate Paths

Update `SSLCertificateFile` and `SSLCertificateKeyFile` to point to the CA signed certificate and private server key, respectively:

```
SSLCertificateFile conf/ssl.crt/server.crt
SSLCertificateKeyFile conf/ssl.key/server.key
```

On Mac OS X, update the the lines to point to the CA signed certificate and private server key, respectively:

```
SSLCertificateFile /certs/server.crt
SSLCertificateKeyFile /certs/server.key
```

VirtualHost (Mac OS X only)

The `VirtualHost` block for `127.0.0.1:443` should be changed to

```
<VirtualHost *:443>
```

Comment out the `ServerName` and `ServerAdmin` lines.

Adjust accordingly if Apache should listen on a different port or only on a specific IP address of a multihoming setup.

The VirtualHost block for 127.0.0.1:80 may be removed or modified as needed to serve unencrypted traffic (such as distributing the CA certificate of an in-house CA).

Launching Apache (Mac OS X)

Apache can be launched from a terminal window as well as from System Preferences. **In some situations Apache must be launched from a terminal window instead of System Preferences:**

- Apache and MassTransit are not running as the same user. (This configuration also requires Mac OS X 10.3).
- An SSL password needs to be entered on startup.

- There are syntax errors in httpd.conf.

The commands to start and stop Apache, which must be run as `root`, are `apachectl start` and `apachectl stop`. **Panther users with Fast User Switching enabled must switch to the user running MassTransit before launching Apache.**

To control Apache via the System Preferences follow these steps:

1. Open System Preferences
2. Open Sharing, under Internet & Network
3. Select "Personal Web Sharing" and click Start.

Apache must be restarted whenever `httpd.conf` is changed.

Launching Apache (Windows)

1. Open a Command Prompt[3] and navigate to the Apache bin directory (typically `C:\Program files\Apache Group\Apache\bin`)
2. To install Apache as a service and launch it, type the following:
   `apache -k install`
   `apache -k Config -D SSL` (SSL installations only)
   `apache -k start`

   To launch Apache once without installing as a service, type the following:

   `apache` (non-SSL installations only)

   `apache -D SSL` (SSL installations only)

   Note: For Apache 1.3 apache.exe may be located in Apache Group\Apache above the \bin directory.
3. If applicable, twice enter the passphrase for the private server key.

The Apache installation program should have installed the Apache Monitor, which appears as a small pink feather in the system tray. From this point forward it is possible to click on the Apache monitor to start, stop, or restart the Apache server.

Troubleshooting Apache

Symptom: When launched from the Mac OS X GUI, Apache never completes startup.

Possible solution: There might be a syntax error in `httpd.conf` or another of Apache's config files. Try launching Apache from a terminal window (see "Launching Apache (Mac OS X)". You should get an error message indicating exactly what line is preventing Apache from starting.

[3] A command prompt window can be opened by clicking on the Start Menu, going to Run…, typing cmd.exe, and pressing Enter.

Symptom: Internal Server Error is displayed in the web browser when a user tries to log in.

Possible solution: This indicates the CGI cannot communicate with MassTransit. One common cause of this is that Apache was launched using the GUI in a scenario where it must be launched from a terminal window. See "Launching Apache (Mac OS X)".

Another common cause is that in `mtadmin.cfg` `APACHE_MODE` is not set to `TRUE`.

Apache httpd for Mac OS X Server

The information in this section applies only to the Apache web server under Mac OS X Server 10.3. **If you are running Mac OS X Server 10.2, please refer to "Apache httpd (including Mac OS X Personal Web Sharing)".** The default web server on Mac OS X Server is implemented with Apache and is largely configurable through a graphical user interface.

Before continuing, complete:

- "User Considerations (Mac OS X)"
- "Configuring the CGI"
- "Configuring MassTransit."
- If SSL is to be used, also complete "Using the MassTransit CSR Tool" and obtain a CA signature, as directed by that section.

Configuring Apache for Mac OS X Server - Basic Server Settings

Most Apache settings can be configured in the Web Server section of the Server Admin user interface.

- Open Server Admin from /Applications/Utilities
- Click on the triangle to expand the list of services.
- Click on the "Web" service.
- Click the "Settings" button at the bottom of the window.

You will need to disable the default site so it can be replaced with the MassTransit web site. Then you will need to configure the new MassTransit site and enable CGI execution. Follow these instructions:

Disable the Default Site

1. Click on the "Sites" button.
2. Double-click the existing (unnamed) site.
3. Change the domain name to "Original" and click "Save."
4. Click the small arrow in the upper right corner of the pane to go back to the sites list.
5. Uncheck the "Original" site to disable it.

Create a New Default Site for MassTransit

6. Click on the "+" button in the Sites pane to add a new site.
7. Set the domain name to the name of your domain (www.yourdomain.com), or leave the field blank.
8. Set that the IP field to the IP address you want the web server to listen on, or "Any" to listen on all interfaces.
9. Change the web folder to the MassTransit Remote Admin folder.
10. Add "default.html" to the list of default index files.
11. Change the administrator email to the appropriate email@your.domain.com.
12. Click the "Options" tab at the top of the pane.
13. Check "CGI Execution."
14. Click "Save" to save the site.
15. Click the small arrow in the upper right to return to general web configuration.
16. Check the box to enable the new site you just configured (the name will be blank).

Enable CGI execution

17. Click the "MIME types" tab at the top of the pane.
18. Click suffix field of the `cgi-script` entry in the content handlers section.
19. Change the suffix field to `cgi,acgi`.

Configuring Apache for Mac OS X Server - httpd.conf

A few of the Apache settings can only be configured in a text configuration file, httpd.conf. Use a text editor to open `httpd.conf`, (in `/etc/httpd/`). On Mac OS X Server, you will need to become root to edit this file (or change the permissions on the file so you can write to it). Make sure you edit the file with an editor that supports Unix-style line endings. Adjust the following settings within httpd.conf:

User and Group setting (Mac OS X only)

If MassTransit will run as `root`, User and Group can be left as:

```
User www
Group www
```

Otherwise set as below, replacing `username` with the correct user (the user who MassTransit will run as):

```
User username
Group staff
```

See the "User Considerations (Mac OS X)" section for more information.

Hide *.cfg files

Disable remote viewing of *.cfg files. Change:

```
<Files ~ "^\.([Hh][Tt])">
```

to:

```
<Files ~ "^\.([Hh][Tt])|^\.cfg">
```

Configuring Apache on Mac OS X Server: Changes for SSL installations

To configure Apache to run as a secure web server using SSL, make sure you have completed all the regular instructions. You must also have a valid, signed certificate, key, and certificate authority. See the "Generating SSL Certificates" and "Signing SSL Certificates" sections for information on obtaining these files.

In `mtadmin.cfg` (in the MassTransit Remote Admin directory), set `WEB_SERVER_SECURE = TRUE`

In `mtadmin.cfg` (in the MassTransit Remote Admin directory), `WEB_SERVER_ADDRESS` should be set to the DNS name (or IP address) and port clients will use to access the web server. Change the '80' to '443', since that is the standard port for SSL encrypted https.

1. Open Server Administration for the web server:
    - Open Server Admin from /Applications/Utilities
    - Click on the triangle to expand the list of services.
    - Click on the "Web" service.
    - Click the "Settings" button at the bottom of the window.
2. Click on the Modules tab.
3. Verify that `ssl_module` is enabled.
4. Click on the Sites tab.
5. Double-click the site you are configuring for MassTransit to edit it.
6. Verify that you have a domain name entered. If you are not using DNS, you can enter anything in this field, but it must not be left blank.
7. Change the port number to 443.
8. Click on the Options tab.
9. Uncheck "Performance Caching."
10. Click on the Security tab.
11. Check the "Enable Secure Sockets Layer (SSL)"
12. Set the Certificate File, Key File, and CA File fields to point to your files. You can also click the edit button for each field and simply paste in the text from your files.
13. Enter the passphrase for the private key in the Pass Phrase field.
14. Save the configuration. You do not need restart the web server (you will being doing this manually).
15. Stop the web server in the UI if it is running.

Launching Apache on Mac OS X Server

If you are **not running MassTransit as `root` and you are not using SSL for your web server**, launch Apache via the Server Settings follow these steps:

1. Open Server Settings (Located in Applications/Utilities)
2. Click on the "Internet" tab
3. Click on the "Web" icon
4. Click "Start Web Server"

Apache must be restarted whenever server settings are changed in the user interface or in the `httpd.conf` file.

If you **are running MassTransit as `root` or you are using SSL for your web server,** you will need to launch Apache from the terminal window following these steps:

1. Open Terminal (Located in Applications/Utilities)
2. Type `apachectl start` at the prompt.

To stop Apache from the terminal, type `apachectl stop` at the prompt. Note that for either command to work, you will need to be logged in as root.

Troubleshooting Apache

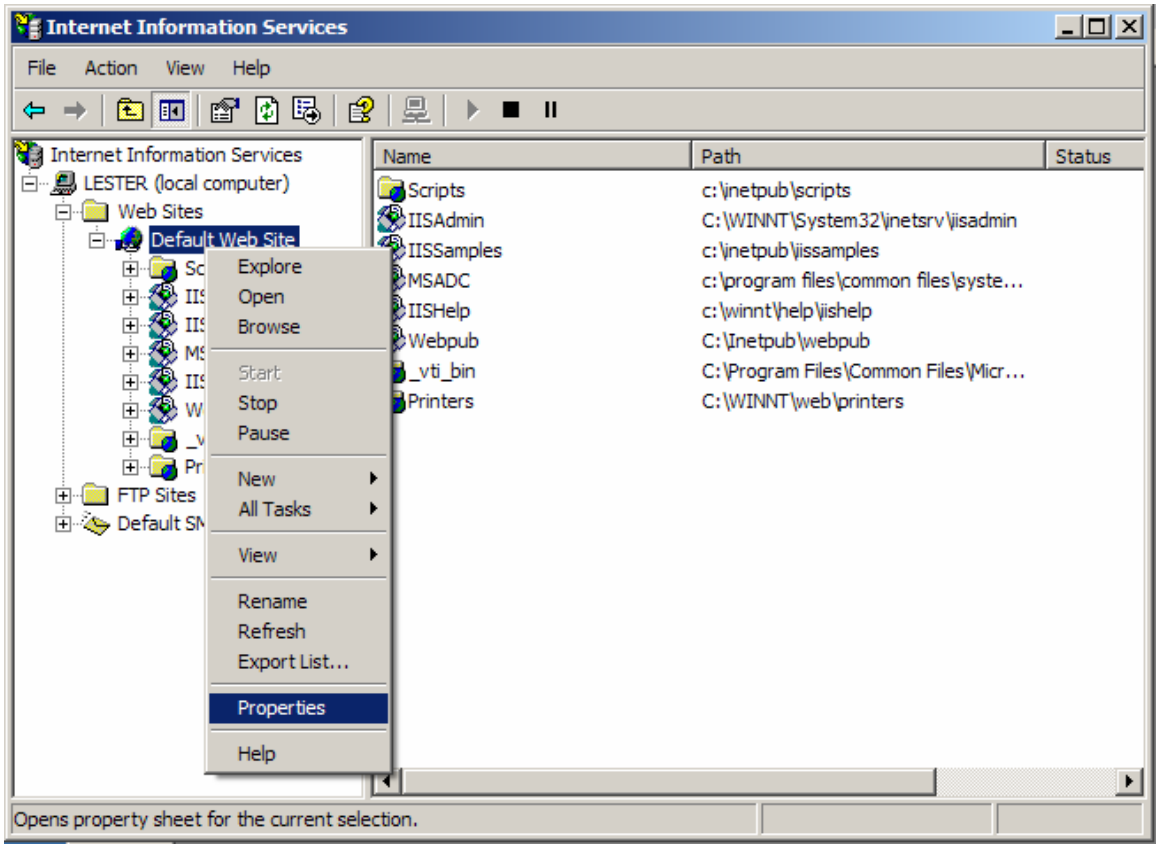
Please see the troubleshooting tips in the main Apache httpd section as well as the general troubleshooting section at the end of this document.

Microsoft Internet Information Services (IIS)

The information in this section applies only to the IIS web server.

IIS is administered via Control Panel → Administrative Tools → Internet Information Services (IIS). If IIS is not listed under the Administrative Tools it can be added via the Add or Remove Programs Control Panel under Add/Remove Windows Components.

Open the properties for the default web site with a right-click as displayed in the screen shot below.



1. Complete all configuration described in the sections “Configuring the CGI” and “Configuring MassTransit”
2. Set `Local Path` to the MassTransit Remote Admin folder in the `Home Directory` tab of the Properties for Default Web Site.
3. Verify that the following default settings have not be changed:
    a. `Read` permission is granted, `Script source access`, `Write`, and `Directory browsing` are not.
    b. `Execute Permissions` are set to `Scripts and Executables`.
    c. `Application Protection` is set to `Medium (Pooled)`
4. On the `Documents` tab, add `default.html` and move it to the top of the list.
5. For multihoming setups, choose the IP Address IIS should use on the `Web Site` tab. IIS’ “socket pooling” feature prevents this setting from working properly and must be disabled. See the Microsoft Knowledge Base Article 259349 at http://support.microsoft.com/default.aspx?scid=KB;en-us;q259349 for more information.
6. On Windows 2003, CGIs must be explicitly enabled. To enable them:
    a. Click on `Web Services Extensions` for the local machine.
    b. Select `All Unknown CGI Extensions` from the details pane.
    c. Click `Allow` to enable CGI execution.

7. IIS is now fully set up for unencrypted http communication. Press the start button to launch it.

Microsoft IIS SSL Configuration

To set up IIS for SSL communication, the following additional steps are required.

As described in the SSL Overview section, a private server key and CSR must be generated. IIS has a built in CSR tool that takes care of both of these steps.

1. In the Properties for Default Web Site, go to the `Directory Security` tab and click `Server Certificate…`. This launches the Web Server Certificate Wizard; click `Next`.
2. Select "Prepare the Request now, but send it later" and click `Next` to fill out the request. Note that:
    a. `Name` is used locally when multiple websites are running on the same machine to distinguish which website the certificate belongs to. This property is not part of the certificate itself.
    b. `Bit length` should be set to the desired encryption level. 1024 is standard; 2048 is becoming increasingly popular.
    c. `Common Name` must be set to the URL clients will use in their browsers.

This completes the CSR generation process. By default, the CSR is stored in c:\certreq.txt. It is now necessary to have a CA sign the CSR. IIS does not accept self-signed certificates. See the "OpenSSL" section for more information on obtaining CA signatures.

3. Before importing the new server certificate into IIS, ensure the CA's certificate has been installed into Windows. Windows comes with many Root CA certificates pre-installed but in-house and lesser known 3rd party CA certificates must be installed by hand. Note that the CA certificate is distinct from the server certificate (which is the CA-signed CSR); both are required.
    a. Give the CA's certificate a `.crt` or `.cer` extension and double click on it. Choose `Install Certificate` and `Auto Choose` the certificate store. The certificate will be stored as a "Trusted Root CA." The error message "This file is invalid for use as the following: Security Certificate" indicates that Windows cannot parse the PEM certificate. See the "OpenSSL" section for information on Window PEM certificate handling.
4. Return to the IIS configuration (via Control Panel → Administrative Tools → Internet Information Services (IIS), properties for Default Web Site), start the Web Server Certificate Wizard (select the `Directory Security` tab and click `Server Certificate…`) and import the signed server certificate.
5. IIS is now ready to accept security https traffic on port 443.
    a. To disable unencrypted traffic, click the `Edit…` button and enable `Require secure channel (SSL)`. Optionally, disable low security encryption by selecting `Require 128-bit encryption` as well.

b. Note that clients must have access to the CA certificate that was used to sign the server certificate. See the Distributing CA Certificates Appendix for more information.

Troubleshooting ISS

When log in is submitted, browser displays "The page cannot be displayed" and/or "HTTP 405 – Resource not allowed"

This indicates that the CGI cannot be executed. Verify the `Execute Permissions` setting was properly set and, if applicable, that the Windows 2003 specific CGI steps were followed.

4D WebSTAR

The information in this section applies only to the WebSTAR web server (version 5.x).

WebSTAR Configuration

1. Complete all configuration described in the sections
   a. "User Considerations (Mac OS X)"
   b. "Configuring the CGI"
   c. "Configuring MassTransit"
2. Install WebSTAR from the installer, following the instructions provided by 4D. Note: based on the user WebSTAR should run as (see "User Considerations (Mac OS X))", a user named `webstar` either must or must not exist on the system. Configure this correctly before continuing.
3. Launch WebSTAR.
4. Open the WebSTAR Admin Client and connect to the running web server.
5. Under `Web Server -> Web Hosts` click on the `DefaultSite`. Select the textbox labeled `Root Folder` and click the associated `Choose…` button. Select the MassTransit Remote Admin folder.

   If desired, specify a more descriptive `Friendly Name` (such as `MassTransit Remote Admin`). This is for administrative purposes only and will be visible to users.

   Note for multihoming setups: Leave existing hosts alone and create a new one for MassTransit Remote Admin. Configure as described above, then add the MassTransit Remote Admin host to the routing list and ensure that for some hostname/IP/port configuration it is the first matching host in the list. For example, MassTransit Remote Admin might be at the top of the list and set to `masstransit.myserver.com:80`, while another virtual host below it is set to `*:80`. Be sure that the IP and port combination used by the MassTransit Server it not in use by WebSTAR. For example, MassTransit Remote Admin might listen on 209.183.196.38:443, leaving 209.183.196.39:443 free for the MassTransit Server.

Click Save.

6. On the menu bar, go to File -> Switch Settings Group and select the site created in step 5.
7. Go to Web Host -> CGI Settings and verify that Allow Unix CGI and FastCGI execution as well as Allow execution from any folder in this virtual host are both checked. If changes were required, click Save.
8. Verify under Web Server -> Default Documents that default.html is listed. If changes were required, click Save.
9. Verify under Web Server -> Web Connections that WebSTAR is set to serve on the port set in mtadmin.cfg (See "Configuring the CGI"). Remove other ports if desired. Click Save.
10. Close WebSTAR Admin.
11. Close and restart the WebSTAR server. WebSTAR is now ready to serve unencrypted http traffic.

4D WebSTAR SSL Configuration

More detailed information about configuring WebSTAR for SSL is provided in the *WebSTAR Technical Reference* document that ships with WebSTAR. As described above in the SSL Overview section, a private server key and CSR must be generated. The WebSTAR Admin Client can do both of these. In the Admin Client menu bar, select Tools -> Generate Keys and CSRs.

1. Enter a name for the private key to be generated as well as a passphrase and key size. 1024 bits is standard; 2048 is becoming increasingly popular.
2. Click Generate Key to create the private key file. It will be stored in the WebSTAR SSLTools/Keys directory.
3. Fill in the fields for the Generate Certificate Signing Request section
4. In the Key dropdown, select the private key generated in steps 1 and 2 and enter the passphrase.
5. Select either Normal Certificate or Self Signed Certificate. See the SSL Overview section for a description of the security implications of using self-signed certificates.
6. Click the Generate CSR button.
7. Select and copy the text that appears in the output area.
    a. If you are using a well known CA such as Verisign or Thawte, submit the CSR via their web forms. Proceed to the next step once you have received your certificate.
    b. If you are using an in-house OpenSSL CA, save the text as a text file and follow the steps in OpenSSL Signing Certificate Requests section. Proceed to the next step once you have received your certificate.

c. If you select self-signed in step 5, save the text as a text file; it is a complete certificate and is already signed.

8. Place your certificate in the WebSTAR `SSLTools/ServerCerts` directory.
9. In the Admin Client, go to `Web Server -> SSL Configurations` and click `New`.
10. Give the configuration a name, select the key generated in steps 1 and 2, and enter the passphrase.
11. Select the certificate from the pull-down. If it isn't listed, try restarting the Admin Client.
12. Select which ciphers to enable. They are listed in decreasing security; a good discussion of them is provided in the WebSTAR Technical Reference. Click Save.
13. Go to `Web Server -> Web Connections` and select the listener for MassTransit.
14. On the `SSL Config` dropdown, select the configuration created in steps 10 through 12. Click `Save`.
15. Close WebSTAR Admin and restart the WebSTAR server. WebSTAR is now ready to serve encrypted https traffic.

SSL Certificate and Key Generation

In order to use SSL to secure your web server, you must obtain or generate a signed SSL certificate and key. The instructions below describe how to generate keys and certificate requests. Once you have a key and certificate request, see the "Signing SSL Certificates" section for details on how to obtain a signed certificate using your request.

Generating SSL Certificates on Windows and Mac OS 9

On Mac OS 9 and Windows, certificates can be generated using the MassTransit CSR tool. On Mac OS X or Mac OS X Server, follow the instructions in the "Generating and Configuring SSL Certificates on Mac OS X" section.

The MassTransit CSR tool simplifies the creation of a CSR (certificate signing request). Use the MassTransit CSR Tool in order to:

- Give the MassTransit server a certificate with which to prove its identity to other MassTransit servers
- Generate a private server key and CSR for the web server to prove its identity to web clients if the web server being used does not have its own CSR tool. Apache (including Mac OS X Personal Web Sharing) does not have its own CSR tool but IIS and WebSTAR do.

To use the CSR Tool, follow these steps:

1. Open the CSR Generator
    a. On Macintosh, open the Security:CSR Generator folder inside the MassTransit folder. Double click on the CSR Generator.

b. On Windows, from a Command Prompt, launch `CSR.EXE` (located in `Security/CSR Generator`)

2. Choose a passphrase for the certificate, and enter the information requested.
    a. COMMON NAME should be the address of the MassTransit server. I.e., `www.some-organization.org`, or `192.168.1.14`. Using a domain name instead of a raw IP address is preferable.
    b. The Organizational Unit Name and 'extra' attributes are optional
3. The CSR Generator will create two files in the CSR directory: `mt_private_key.pem` and `mt_cert_req.pem`. These files are the server private key and CSR, respectively.
4. Obtain a signature to convert the CSR into a certificate. For information on signing CSRs, see the "Signing SSL Certificates" section.

Once signed, the certificate and private key file can be used by MassTransit or a web server.

Generating SSL Certificates on Mac OS X

The information in this section only applies if SSL is to be used on Mac OS X or Mac OS X Server. These instructions describe how to generate a server key and certificate request for and SSL certificate. Note that if you are using WebSTAR, you should probably use the certificate generation tool packaged with WebSTAR.

1. Make sure you are logged in as an administrator.
2. Open a terminal window.
3. Type `sudo mkdir -p /certs`
4. Enter your password.
5. Type `cd /certs`
6. Type openssl genrsa –des3 –out server.key 1024 to generate a new private key for your certificate.
7. When prompted, enter a new passphrase for your server key.
8. Type openssl req –new –key server.key –out server.csr to generate a certificate request.
9. Enter the passphrase for the server key that you selected in step 7.
10. Answer the series of questions when prompted. Your responses are not critical *except* for "Common Name." The common name you enter must be the fully qualified name of your web server, like www.grouplogic.com. Leave the challenge password blank.

You now have a valid server key and certificate request. See the "Signing SSL Certificates" section for information on how to obtain a signed certificate using your certificate request.

Signing SSL Certificates

Certificate requests must be signed before they can be used for authentication on a secure web server. In general, certificate requests should be sent to a recognized certificate authority, like Verisign, to obtain a signed certificate. However, you may generate a self-signed certificate locally instead.

Before self-signing a CSR, be sure to understand the security implications discussed in the section "SSL Overview". This certificate will provide no guarantee that its holder is the person named in the certificate. Many web servers will refuse to serve self-signed certificates.

Self-Signing SSL Certificates on Windows

OpenSSL is an open-source SSL toolkit that can perform the various certificate related tasks required to set up SSL. It is also used internally by MassTransit and Apache to perform the actual encryption / decryption of traffic. The official OpenSSL homepage is http://www.openssl.org. Windows x86 binary distributions are available at http://hunter.campbus.com.

Installing OpenSSL

1. Download (or compile) an OpenSSL binary and unpack it into its own directory. These instructions assume OpenSSL has been placed in `c:\Program Files\OpenSSL`.
2. OpenSSL requires a configuration file. The online OpenSSL documentation provides sample configuration snippets. The “sample configuration file prompting for field values” at http://www.openssl.org/docs/apps/req.html combined with “sample configuration file with the relevant sections for ca” at http://www.openssl.org/docs/apps/ca.html will do. Combine these snippets into a file named openssl.cfg located in the OpenSSL directory. Remove the line:

   The resulting file should look like this:

```
[ req ]
default_bits           = 1024
default_keyfile        = privkey.pem
distinguished_name     = req_distinguished_name
attributes             = req_attributes
x509_extensions        = v3_ca

dirstring_type = nobmp

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = AU
countryName_min                 = 2
countryName_max                 = 2

localityName                    = Locality Name (eg, city)

organizationalUnitName          = Organizational Unit Name (eg, section)

commonName                      = Common Name (eg, YOUR name)
commonName_max                  = 64
```

```
emailAddress                    = Email Address
emailAddress_max                = 40

[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 4
challengePassword_max           = 20

[ v3_ca ]

subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid:always,issuer:always
basicConstraints = CA:true


[ ca ]
default_ca      = CA_default            # The default ca section

[ CA_default ]

dir             = ./demoCA              # top dir
database        = $dir/index.txt        # index file.
new_certs_dir   = $dir/newcerts         # new certs dir

certificate     = $dir/cacert.pem       # The CA cert
serial          = $dir/serial           # serial no file
private_key     = $dir/private/cakey.pem# CA private key
RANDFILE        = $dir/private/.rand    # random number file

default_days    = 365                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = md5                   # md to use

policy          = policy_any            # default policy
email_in_dn     = no                    # Don't add the email into cert DN

nameopt         = ca_default            # Subject name display option
certopt         = ca_default            # Certificate display option
copy_extensions = none                  # Don't copy extensions from request

[ policy_any ]
countryName             = supplied
stateOrProvinceName     = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
```

3. Point OpenSSL to the configuration file by setting the `OPENSSL_CONF` environment variable[4]:

```
set OPENSSL_CONF=c:/Program Files/OpenSSL/openssl.cfg
```

[4] It is non-standard to specify quoted paths in an environment variable but in some situations this is required to make OpenSSL work with paths that have spaces. If OpenSSL displays "error loading the config file" try putting double quotes around the path. Environment variables may be set temporarily for a single command prompt window by using the set command or permanently in the Advanced tab of the System Control Panel.

Signing Certificate Requests

Once OpenSSL is installed, you can use it as a CA (applies to any web server using an in-house CA) take the following steps:

1. Create a directory structure for the CA to store its files.
    a. On Windows, open a command prompt and go to C:\Program Files\OpenSSL\. On Mac OS X, open a terminal window and go to a directory of your choice. This directory will be referred to as the "OpenSSL root directory".
    b. Create a directory structure for the CA data to reside in. Inside the OpenSSL root directory, create a directory named demoCA. Inside the demoCA directory, create the following subdirectories: certs, newcerts, crl, and private.
    c. Inside the demoCA folder, create a completely empty file named index.txt and a file named serial that contains 01 followed by a blank line.
2. Create a CA Certificate
    d. Go to the OpenSSL root directory5.
    e. Enter the following command:

```
openssl req -newkey rsa -x509 -days 365 -outform PEM -out
demoCA/cacert.pem -keyout demoCA/private/cakey.pem
```

    f. Supply information as prompted.
5. The CA certificate `cacert.pem` created in the the `demoCA` folder must be distributed to users. The private key should not be distributed! See the Distributing CA Certificates Appendix for more information.

The CA is now ready to sign CSRs.

Some web servers will not accept server certificates that have been signed by an unknown CA. In this case, the instructions above for the web server being used will include information on how to import the CA certificate. However, the following steps will be necessary before the CA certificate can be imported:

Windows users: Some PEM certificates must be edited in a text editor before Windows can use them. The core of a PEM certificate is the base64 encoding of the certificate data enclosed in a `-----BEGIN CERTIFICATE----- -----END CERTIFICATE-----` block. Windows will not recognize PEM certificates that include any text outside of this block. Simply remove all text outside the block and save the file.

Cross-platform use: PEM files are text files and, as such, suffer from line-ending incompatibilities when moved from one platform to another. If you generate a PEM file on one platform and wish to use it on another, use a text editor or utility to

[5] OpenSSL can be run from other locations if the paths in `openssl.cfg` are changed to be absolute instead of relative. On Windows the OpenSSL directory must also be added to the `PATH`. Mac OS X's `openssl.cfg` is located in `/System/Library/OpenSSL`.

convert the line-endings to the platforms native format. UNIX systems including Mac OS X use line feeds (\n or 0x0a), Mac OS 9 and earlier use carriage returns (\r or 0x0d), and Windows machines use both (\r\n or 0x0a 0d).

Options for Signing Certificate Requests

Self-Signing a CSR

Before self-signing a CSR, be sure to understand the security implications discussed in the section "SSL Overview". This certificate will provide no guarantee that its holder is the person named in the certificate. Many web servers will refuse to serve self-signed certificates.

```
openssl x509 -in "/Program Files/MassTransit/Security/CSR
Generator/mt_cert_req.pem" -out "/Program
Files/apache2/conf/server.crt" -req -signkey
mt_private_key.key -days 31
```

Signing requests with a CA signature

Follow the steps in "Creating a CA Certificate" above before attempting to sign requests with a CA signature. Adjust the command below to reflect the location where the CSR is and run it from the OpenSSL root directory. The syntax below matches IIS, which by default places a CSR named `certreq.txt` in C:\.

```
openssl ca -in /certreq.txt -out /server.crt -days 365
```

Additional OpenSSL Tasks

Removing the passphrase from (decrypting) a private key for use with Apache

Before decrypting a private key, be sure to understand the security implications discussed in the subsection "Obtaining an SSL Certificate" of the Apache section. If an unauthorized person obtains access to a decrypted private key, the key and all certificates signed with it are compromised and must be revoked.

```
openssl rsa -in mt_privkey.pem -out server.key
```

Converting between PEM and DER format for distribution to clients

PEM is the default format on most platforms. PEM consists of the DER format base64 encoded with additional header and footer lines. Thus PEM files look like alphanumeric data contained inside a `-----BEGIN CERTIFICATE----- -----END CERTIFICATE-----` block whereas DER certificates are raw binary data.

Converting certificates

```
openssl x509 -in cert.pem -inform PEM -out cert.der -
outform DER
```

```
openssl x509 -in cert.der -inform DER -out cert.pem -
outform PEM
```

Converting keys

```
openssl rsa -in key.pem -outform DER -out keyout.der
openssl rsa -in key.der -outform PEM -out keyout.pem
```

Self-Signing SSL Certificates on Mac OS X

Use the following instructions to sign a certificate you have generated. Note that these instructions assume you have already followed the instructions in "Generating SSL Certificates for Mac OS X."

1. Open a Terminal window.
2. Type `cd /certs` to go to your certificates directory.
3. Type `openssl genrsa -des3 -out ca.key 2048` to generate the key for your certificate authority.
4. When prompted, enter a new passphrase for your certificate authority (CA).
5. Type `openssl req -new -x509 -days 4096 -key ca.key -out ca.crt` to generate your CA.
6. Enter the passphrase for your CA that you selected in step 4.
7. Type the following series of commands to generate the folder hierarchy required to sign the certificate:

```
mkdir -p demoCA/private
cp ca.key demoCA/private/cakey.pem
mkdir demoCA/newcerts
touch demoCA/index.txt
echo "01" > demoCA/serial
```

8. Type `openssl ca -policy policy_anything -in server.csr -out server.crt` to sign your certificate with your CA.
9. When prompted, enter the passphrase for your CA that you selected in step 4.

Now you have a self-signed certificate. The important files you will need to configure your web server are now in the "certs" directory on your local volume:

Key File – `server.key`

Certificate File – `server.crt`

CA File – `ca.crt`

Distributing CA Certificates

A CA signed certificate has no meaning if the signature is not guaranteed to belong to CA. For this reason it is imperative to distribute CA certificates in a way that guarantees their authenticity. Most SSL enabled applications, including MassTransit and the web browsers it supports, come with CA certificates for known Root CAs pre-installed. Therefore only in-house and lesser known 3rd party CA certificates need to be

distributed. Note: server certificates do not need to be explicitly delivered; they are communicated automatically during SSL negotiation.

CA certificates may simply be copied to disk and hand-delivered, but they can also be electronically delivered. The fingerprint of the certificate should always be verified to ensure security, but this is especially important for electronic distribution.

To determine the fingerprint of the certificate:

1. In Windows, it is possible to just double click on a .crt or .cer file and read the fingerprint in the details tab of the window that opens.
2. Alternatively, OpenSSL can be used to display the fingerprint of a certicate (commands for PEM and DER certificates, respectively):

```
openssl x509 -noout -fingerprint -in server.crt
openssl x509 -noout -fingerprint -inform DER -in
server.crt
```

Distributing CA Certificates to Web Browsers

The simplest method of distributing a CA certificate to a client web browser is to distribute the certificate with a web server.

To distribute the certificate via a web server do the following:

1. Place the certificate in a directory being served by the web server (the "MassTransit Remote Admin" directory will work)
    a. For older versions of IE, including IE 5.1 Mac, the CA certificate must be distributed in DER format. See "Converting between PEM and DEM format." Only the CA certificate has to be DER format; server certificate may be in PEM format. Place both the DER and PEM format certificates on the server to accommodate all browsers.
2. Load the certificate in a web browser on the machine which is to receive the certificate. I.e., go to http://mastransit.some-organization.org/server.crt
    b. Choose "View Certificate" or "Open" and "Details" depending on what dialog is presented in the browser.
3. Compare the fingerprint shown with the known value.

Distributing CA Certificates to MassTransit Servers

1. Send the CA certificate via MassTransit just like any other file.
2. The recipient should use one of the methods described above to determine the fingerprint of the received certificate.
3. Compare the fingerprint shown with the known value.

Opening Terminal and Command Windows

On Mac OS X, a terminal window can be opened by running the Terminal application located in the `/Applications/Utilities` folder.

On Windows, a command prompt window can be opened by clicking on the Start Menu, going to `Run…`, typing `cmd.exe`, and pressing Enter.

Enabling the Root Account on Mac OS X

1. Open the NetInfo app in `/Applications/Utilities/NetInfo Manager`
2. Go to the security menu and select `"Enable root user"`. You will be prompted to enter a password.
3. Go to `System Preferences -> Accounts`
4. Switch to the `Login Options` tab
5. Select `"Display Login Window as: Name and password"`
6. Log out of your current user
7. The login screen should now allow you to type a username (`root`) and the password you selected.

Note that most programs (including MassTransit) installed as root will have unexpected results when run by other users. It is therefore best to not use root to install programs unless specifically instructed to do so.

Running Commands as Root (and others) on Mac OS X

In a Mac OS X terminal window, the command `su username` can be used to switch to a different user. You will be prompted for a password. Type `exit` to return to the original user. If you are not currently logged in as an administrator, you may have to first `su` to an administrator before being able to `su` to root.

Converting Line Endings

Mac OS 9 indicates the end of a line with a carriage return (often written as CR, \r, or 0x0d). Mac OS X, like other UNIX operating systems, uses line feeds (often written LF, \n, or 0x0a). Windows (as well as DOS) uses both back to back (often written CRLF, \r\n, or 0x0d0a). Many text editors, such as BBEdit and TextPad, can convert line endings from one format to another.

From a Mac OS X terminal window the line endings of a file can easily be converted using perl. The following command converts `filename` from OS 9 carriage returns to OS X line feeds:

```
perl -pi -e 's/\r/\n/g' filename
```

Troubleshooting

Can't find a server. Code: 42 is displayed in the web browser when a user tries to log in.

This error indicates that MassTransit is not running; launch it and try again.

Plug-in tab does not display but the rest of remote admin interface works.

This problem may occur if the IP address is not correctly set in `mtadmin.cfg`.

Clients can log in but cannot transfer files

In this case, the client can communicate successfully with the web server and the CGI is successfully communicating with the MassTransit server, but the MassTransit Assistant running alongside the client's web browser cannot communicate directly to MassTransit. Verify that no firewalls are blocking traffic and that the MassTransit server is listening for traffic. The MassTransit server must listen on a port that is not in use by the web server.