

MQSeries Everyplace – WTLS Mini-Certificate Server User Guide: v1.00

SupportPac Category 3 – September 2002

Barry Aldred
Mike Cobbett
IBM Corporation
Hursley Park
Winchester
UK
SO21 2JN

Take Note!

Before using this report be sure to read the general information under "Appendix C: Notices" on page 83.

License warning

MQSeries Everyplace – WTLS Mini-Certificate Server version 1.00 is supplied under the terms of the International Program License Agreement, a copy of which is reproduced in "Appendix D" on page 85. Defect correction for this release will be provided under that agreement for users holding valid MQSeries Everyplace deployment license(s) until the end of service date, June 30, 2003.

Please refer to <http://www.ibm.com/software/mqseries> for details of the license conditions pertaining to the MQSeries Everyplace product.

First Edition, September 2002

This edition applies to version 1.00 of *MQSeries Everyplace for Multiplatforms – WTLS MiniCertificate Server* and to all subsequent releases and modifications unless otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2002.** All rights reserved.
Note to US Government Users -- Documentation related to restricted rights -- Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule contract with IBM Corp.

Table of contents

Figures	v
Summary of amendments	vi
Preface	vii
*** Upgrade notice ***	vii
Bibliography	viii
Related material	viii
Web references.....	ix
Download sites.....	ix
Newsgroups.....	ix
Standards	ix
Tcl reference information	ix
1 Introduction	1
1.1 Further reading	1
2 Installation.....	2
2.1 Environments.....	2
2.2 Contents	3
2.3 Installation.....	4
Preparation	4
MCS – graphical user interface version (Windows only)	5
MCS – command line version (all platforms).....	6
3 Concepts.....	7
3.1 Overview.....	7
3.2 Authenticatable entities	7
3.3 Mini-certificates	9
3.4 Registry.....	9
3.5 Profiles.....	10
3.6 Operator interfaces to MCS	11
4 Getting started	12
4.1 Loading a profile and creating entities.....	13
Using MQe_MiniCertServer	13
Using CommandConsole	21
4.2 Creating a secure queue manager with mini-certificates.....	23
4.3 Log and summary inspection	27
Using the MQe_MiniCertServer	27
Using CommandConsole	30
4.4 Authorizing the issue of a queue certificate.....	31
Using MQe_MiniCertServer	31
Using CommandConsole	31
4.5 Deleting an authenticatable entity	31
Using MQe_MiniCertServer	31
Using CommandConsole	32
5 MQe_MiniCertServer: graphical user interface	33
5.1 Main window	33
Layout.....	33
Navigation and control	34
Menus.....	34
List view pane	48
5.2 Report window	49
Menus	50
5.3 Trace window.....	51
Menus.....	52
6 CommandConsole: text command user interface.....	56
6.1 Outline	56
Objective.....	56
General approach	56
Symbols, notation and formatting of examples.....	56
Resources manipulated	57
Sources of information on Tcl and Jacl	57

	Starting up the CommandConsole program	57
	Use of parameters when starting the CommandConsole program	58
	Stopping the CommandConsole program	58
	Loading and using the mqe_cert package from within a separate Jacl shell	58
6.2	Command reference	60
	mqe_cert_help command	60
	mqe_cert_profile command	61
	mqe_cert_server command	65
	mqe_cert_auditlog command	68
	mqe_cert_trace command	70
	mqe_cert_entity command	72
6.3	Scripting using Tcl and the mqe_* commands	76
7	Appendix A: Reference section	78
7.1	Icons and buttons	78
	Icons	78
	Buttons	78
7.2	Accessibility features	80
	Shortcuts	82
7.3	Password and passphrase rules	82
8	Appendix B: Notices	83
	Trademarks	84
9	Appendix C: International Program License Agreement	85
	Part 1 - General terms	85
	Part 2 - Country-unique terms	87

Figures

Figure 4-1: The initial MQe_MiniCertServer window	13
Figure 4-2: MQe_MiniCertServer report window	13
Figure 4-3: Create new profile – General tab	14
Figure 4-4: Create new profile – General tab, with data entered	15
Figure 4-5: Create new profile – Comms tab	16
Figure 4-6: Create new profile – Storage tab	17
Figure 4-7: The running MQe_MiniCertServer window	18
Figure 4-8: The new queue manager entity dialog – General tab	18
Figure 4-9: The new entity dialog – General tab, with data entered	19
Figure 4-10: The new entity dialog – Address tab	20
Figure 4-11: Main window before auto-registration	20
Figure 4-12: Report window before auto-registration	21
Figure 4-13: Create queue manager – General tab	23
Figure 4-14: Create queue manager – Registry tab	24
Figure 4-15: Create queue manager – Security tab	25
Figure 4-16: Queue manager certificate request PIN prompt	25
Figure 4-17: Queue manager certificate details	26
Figure 4-18: Report window after auto-registration	27
Figure 4-19: The MQe_MiniCertServer window after auto-registration	27
Figure 4-20: Modify entity – General tab	28
Figure 4-21: Modify entity – Address tab	29
Figure 4-22: Modify entity – Certificate tab	29
Figure 4-23: The new queue entity dialog – General tab	31
Figure 5-1: An example of the main MQe_MiniCertServer window	33
Figure 5-2: New profile dialog – General tab	35
Figure 5-3: New profile dialog – Comms tab	36
Figure 5-4: New profile dialog – Storage tab	37
Figure 5-5: New entity dialog for a queue manager – General tab	38
Figure 5-6: New entity dialog – Address tab	39
Figure 5-7: Open profile dialog	40
Figure 5-8: Edit profile dialog	41
Figure 5-9: Property entity dialog for a queue – General tab	42
Figure 5-10: Property entity dialog – Address tab	44
Figure 5-11: Property entity dialog – Certificates tab	45
Figure 5-12: System information panel	47
Figure 5-13: The report window	50
Figure 5-14: The trace window	52
Figure 7-1: General keystroke aids to navigation	80
Figure 7-2: Restricted keystroke aids to navigation	81
Figure 7-3: Keystroke shortcuts	82

Summary of amendments

Date	Changes
19 September 2002	Version 1.0 (initial release)

Preface

The MQSeries Everyplace – Mini-Certificate Server is a support program for MQSeries Everyplace (MQe). It issues and renews Wireless Transport Layer Security (WTLS) certificates¹ to MQe queue managers and their associated queues, to enable certificate-based security operations. The program can be run either through a command-line interface on all server platforms, or can be run through a graphical user-interface on Windows server platforms. In either case MQe must support the server platform for high security operation.

*** Upgrade notice ***

If upgrading from the example mini-certificate server shipped with MQe version 1.27 or earlier, please note:

- MQSeries Everyplace version 1.27 is required – you must install that level (or later) before installing this SupportPac.
- Only the WTLS mini-certificates in the standard format are permitted; the earlier format, used by releases of MQe before the WTLS standard was finalized, is not supported.
- The certificate registry used by the example mini-certificate server is not compatible with the registry used by this SupportPac

¹ The WTLS certificate is standardized by the WAP Forum. Further details are available from <http://www.wapforum.org/>. See specifications WAP-211-WAPCert-20010522-a and WAP-211_104-WAPCert-20010928-a.

Bibliography

- *MQSeries Everyplace Version 1.2: Introduction*, IBM Corporation, SC34-5843
- *MQSeries Everyplace Version 1.2: Java Programming Guide*, IBM Corporation, SC34-5845
- *MQSeries Everyplace Version 1.2: Java Programming Reference*, IBM Corporation, SC34-5846
- *Websphere MQ Everyplace SupportPac EC01: MQSeries Everyplace: Configuration Guide*
- *Websphere MQ SupportPac MA88: MQSeries Classes for Java.*

Related material

- *Websphere MQ Everyplace SupportPac EA01: MQSeries Everyplace - XML conversion utility*
- *Websphere MQ Everyplace SupportPac EAP1: MQSeries Everyplace - Device code for Palm OS*
- *Websphere MQ Everyplace SupportPac ED01: MQSeries Everyplace - Get Started*
- *Websphere MQ Everyplace SupportPac ED02: Using MQSeries Everyplace with WebSphere Everyplace Server.*
- *Websphere MQ Everyplace SupportPac EP01: MQSeries Everyplace – Performance report*
- *Websphere MQ Everyplace SupportPac ES02: MQSeries Everyplace – MQe_Explorer*
- *Websphere MQ Integrator SupportPac ID03: MQSeries Integrator – Working with MQSeries Everyplace*
- *Welsh, Brent B : “Practical Programming in Tcl and Tk”, Prentice Hall PTR, ISBN: 0-13-022028-0*

Web references

The following URLs provide useful resources for both MQSeries Everyplace and MQe_Explorer:

Download sites

IBM Websphere (MQSeries SupportPacs):

<http://www.ibm.com/software/ts/mqseries/txppacs/>

IBM Boulder (MQSeries Everyplace downloads):

<http://www6.software.ibm.com/dl/mqsem/mqsem-p>

IBM Visual Age Micro Edition (Java stacks & related technologies):

<http://www.embedded.oti.com>

Microsoft Corp. (JVM downloads):

<http://www.microsoft.com/java/download.htm>

Newsgroups

IBM Software Group (MQSeries Everyplace newsgroup):

<news://news.software.ibm.com/ibm.software.websphere.mqeveryplace>

Standards

WAP Forum Specifications:

<http://www.wapforum.org/what/technical.htm>

Tcl language style guide:

<http://tcl.activestate.com/doc/styleGuide.pdf>

Tcl reference information

The following site is recommended for an introduction to Tcl and other information:

<http://tcl.activestate.com/doc/>

1 Introduction

The MQSeries Everyplace Mini-Certificate Server (MCS) is a member of the MQSeries Everyplace (MQe) family of messaging products. It issues and renews Wireless Transport Layer Security (WTLS) standards-compliant certificates to MQe queue managers (and optionally also to their associated queues) to enable MQe certificate-based security operations. The program can be controlled either through a command-line interface (all server platforms), or can be run through a graphical user-interface (Windows server platforms only). In all cases, MQe must have been previously installed.

MQe queue managers can use WTLS certificates for both queue-level and message-level security, the queue-level functions are provided by the authenticator class *com.ibm.mqe.attributes.MQeWTLS CertAuthenticator*, the message-level functions are provided by the class *com.ibm.mqe.attributes.MqeMTrustAttribute*, both shipped with MQe.

1.1 Further reading

An overview of the principles underlying MQe security are given in the *MQSeries Everyplace: Introduction*; this covers the concepts of authentication, encryption, compression and the MQe provision of queue-based and message-based security.

The *MQSeries Everyplace: Configuration Guide* details the steps required to take advantage of certificate-based authentication. It describes queue manager creation, registry type selection, auto-registration with MCS and certificate issuance and renewal. It gives examples, using the management tool MQe_Explorer or through Java programming, for object creation and administration.

The *MQSeries Everyplace: Java Programming Guide* provides information on the use of message-level security.

The *MQSeries Everyplace: MQe_Explorer User Guide* describes the graphical user interface tool that allows the management of MQe object properties, including authentication and the display, issuance and renewal of certificates.

2 Installation

2.1 Environments

MCS executes as an application running in a Java virtual machine and has dependencies on classes shipped with the MQe product.

MCS version 1.00 requires a local installation of:

- *MQSeries Everyplace version 1.27 or later.*

The MQe product document describes the supported environments. MCS requires that full MQe security class support is available.

Additionally, only if the graphical user interface is to be deployed, the following is also required:

A Microsoft Windows environment² with the following characteristics:

- *Microsoft Java Virtual Machine* (at a level of version 5.00.3155 or later³).
- *Microsoft Foundation Classes* (MFC) – these are always present on the supported Windows operating systems.

Windows 2000 systems meet these requirements, as does Windows NT 4.0 and Windows 98, with the correct level of Microsoft JVM installed. New installations of Windows XP do not include the JVM by default and therefore it must be downloaded from the Microsoft web site (see related footnote below). Upgrades to Windows XP preserve an existing JVM. If a choice of operating system is available, Windows 2000 or XP is to be preferred.

The MCS examples in this book using the graphical user interface assume that the *MQSeries Everyplace: MQe_Explorer v1.27 (or later)* has also been installed⁴. This program is not required for MCS operation but installation is strongly recommended.

Additionally, only if the command-line interface is to be deployed, the following is also required:

A Jacl implementation.

The command line interface executes using Jacl technology, a 100% java implementation of the Tcl scripting language. The Jacl code is not shipped in this supportpac, but is available for download separately from the <http://tcl.activestate.com/software/java/java.html> site. You should download the Jacl package and install the jacl.jar and tcljava.jar files on your CLASSPATH before invoking the tool.

² On Windows 95, 98 and ME there are minor functional deficiencies. Tooltips are not provided – affecting the provision of hover help for toolbar buttons, queue & message object class feedback on node selection in the object tree, etc.

³ To determine the level of the Microsoft JVM type the command JVIEW at a DOS prompt. If the command is not recognized then your machine does not have a Microsoft JVM installed and any other JVM is not acceptable. JVM SDK downloads to install or upgrade are available from <http://www.microsoft.com/java/download.htm>.

⁴ The functions used in the examples are only available in MQe_Explorer v1.27 or later.

2.2 Contents

In addition to this User Guide, the SupportPac includes the files:

- *CommandConsole.jar*
A file containing the MCS classes and resources; it contains all that is required to run MCS from a command line.
- *MQe_MiniCertServer.exe*
An executable that provides a graphical user interface to MCS.

2.3 Installation

The following instruction is important:

Capitalization is significant to both MQe and MCS. At all times, both during installation and subsequent use, ensure that any text input matches the capitalization given in this document – this instruction applies to all names and strings (i.e. class names, alias names, directory names, file paths and names etc). Descriptive text is the only exception.

Preparation

The instructions below assume that the standard MQe v1.27 installation defaults have been adopted⁵. Thus, as a brief check on the MQe installation, confirm that a directory `C:\Program Files\MQe\Java\com\ibm\mqe` containing various `.class` files and other sub-directories (such as `\adapters`, `\administration` etc) exist. Also the directory `C:\Program Files\MQe\Java\examples` should also be present, again with many sub-directories (for example `\administration`). If these directories and files are missing you do not have a complete MQe installation and must re-install.

It is important to ensure that the `classpath`⁶ variable has been set in accordance with the MQe installation instructions.

On Windows systems there is a choice available between either setting the *classpath system variable*, or setting the *classpath user variable*. The system variable (illustrated below) will apply to all users of the machine; the user variable applies only to the selected user. Note that any value present for the user variable will over-ride the system variable value for that user.

On Windows 2000 & XP:

From the *Windows Start* button, go to *Settings*, then *Control Panel* and click on the *System* icon to bring up the *System Properties* panel. Click the *Advanced* tab followed by the *Environment Variables...* button. In the resulting *System Variables* section check the *classpath* definition. It should appear as:

```
classpath      C:\Program Files\MQe\Java
```

Alternatively, the string “C:\Program Files\MQe\Java” should be present in the value.

If *classpath* is not present, click *New...*, then add “*classpath*” in the *Variable Name* input field and “C:\Program Files\MQe\Java” in the *Variable Value* field; click *OK*, then exit the previous panel via the *OK* button.

If *classpath* is present with the wrong value, then select *classpath*, click *Edit...*, and append the string “;C:\Program Files\MQe\Java” to the existing *Variable Value*. Click *OK*, then exit the previous panel via the *OK* button.

⁵ Non-default MQe installations are acceptable if the appropriate name substitutions are made for those given in this document.

⁶ Once MQe_MiniCertServer is running, the value of the *classpath* variable is shown in *System Information*, accessed from the *Help*→*About MQe_MiniCertServer* menu item.

On Windows NT:

From the *Windows Start* button go to *Settings*, then *Control Panel* and click on the *System* icon to bring up the *System Properties* panel. Click the *Environment* tab. In the *System Variables* section check the *classpath* definition. It should appear as:

```
classpath      C:\Program Files\MQe\Java
```

Alternatively, the string “C:\Program Files\MQe\Java” should be present in the value.

If *classpath* is not present, click an item in the *System Variables* section, then add “classpath” in the *Variable* input field and “C:\Program Files\MQe\Java” in the *Value* field; click *SET*, then exit via the *OK* button.

If *classpath* is present with the wrong value, then select *classpath*, and append the string “;C:\Program Files\MQe\Java” to the existing string in the *Value* field. Click *SET*, then exit via the *OK* button.

On Windows 98:

From the *Windows Start* button go to *Run...* In the program name input field type “sysedit” and click *OK*. The *System Configuration Editor* appears; click on the C:\AUTOEXEC.BAT window to bring it to the front. In this file the line:

```
set classpath=C:\Program Files\MQe\Java
```

should be present, or alternatively the string “C:\Program Files\MQe\Java” should be present in the *classpath* value. If *classpath* is missing add the line shown above; if *classpath* is present with the wrong value, append the string “;C:\Program Files\MQe\Java” to the existing value. Go to *File*, click *Save*, then *Exit*.

If the *classpath* system variable is not correctly set MCS (and any other MQe applications) will not be able to locate the MQe classes and will not execute.

MCS – graphical user interface version (Windows only)

On Windows systems only, the **MQe_MiniCertServer** version of MCS may be installed that supports a graphical user interface. The command line version may also be installed on Windows platforms (see below) giving a choice of interface when running on Windows. Before installation of the graphical user interface, check that there is a Microsoft JVM installed on the machine and that it is at the correct level – see the details given in *Environments on page 2*⁷.

To install:

1. Create (or use) a directory named *C:\Program Files\MQe\Java\com\ibm\mqe\mqe_minicertserver* and move the file *MQe_MiniCertServer.exe* into it.
2. Create a desktop shortcut to *MQe_MiniCertServer* by locating the *MQe_MiniCertServer.exe* file in the *Windows Explorer*, right click on it and select *Create Shortcut*. Drag the resulting shortcut to the desktop. Rename if desired.
3. No *classpath* changes are required.

⁷ If on starting *MQe_MiniCertServer* you get the error message “Unable to start the application -- the Java Virtual Machine cannot be loaded. Class not registered”, then the JVM is not at the correct level.

MCS – command line version (all platforms)

On all platforms the **CommandConsole** version of MCS may be installed that supports a command line interface with full scripting capabilities.

To install:

1. Create (or use) a directory named *C:\Program Files\MQe\Java\com\ibm\mqe\mqe_minicertserver* and move the file *CommandConsole.jar* into it.
2. Add *CommandConsole.jar* to the *classpath*.
3. Install Jacl technology on your machine.

The command line interface executes using Jacl technology, a 100% java implementation of the Tcl scripting language. The Jacl code is not shipped in this supportpac, but is available for download separately from the <http://tcl.activestate.com/software/java/java.html> site. You should download the Jacl package and install the *jacl.jar* and *tcljava.jar* files on your *classpath* before invoking the tool.

3 Concepts

3.1 Overview

MCS issues certificates to, and stores information on, *authenticatable entities*. An authenticatable entity is either an MQe queue manager or a queue. An operator supplies information to MCS on the authenticatable entities to be supported and is able to authorize the issue of a certificate. Subsequently, authenticatable entities requests certificates over the network; such requests include a *request PIN*, which is used to identify the requesting entity. Requests are either accepted or rejected by MCS, based on the information held by MCS on that entity.

Authenticatable entities will either request a new certificate or the renewal of an existing certificate. New certificates are issued with unique public and private keys; renewal of an existing certificate retains the current public and private keys.

MCS runs as an MQe application in its own JVM. Unlike most MQe applications, MCS does not instantiate a queue manager, but it does use other common MQe objects; for example, it uses MQe *channels* to communicate with queue managers and an MQe secure *registry* to hold certificates and other information. The registry is protected with a *passphrase*; this passphrase is not stored by MCS and must therefore be supplied by the operator when required.

The MCS also has the concept of a *profile*, used to hold its configuration data. Amongst other information, the profile determines the port and the network adapter to be used for incoming certificate requests. Multiple profiles may be defined, each defining different MCS operational characteristics.

MQe client/server channels are used to connect queue managers to MCS; the MCS acts as an MQe server and listens for incoming channel requests. Once a channel request is received it spawns a socket and uses that new port for continuing data exchange with the remote queue manager; the original port continues to be available for servicing new incoming channel requests. Remote queue managers, acting as clients, use these channels to request (or renew) certificates; the requests are either for queue manager credentials or for queue credentials. MQe queue managers do not use connection definitions to connect to MCS; instead they explicitly identify the port, IP address and the network adapter of the certificate server to be used⁸.

3.2 Authenticatable entities

An authenticatable entity is an MQe object that has its own credentials, i.e. its own certificate and the associated private key. MQe itself creates the following kinds of authenticatable entities⁹:

- Queue managers
- Queues

Other entities may share the credentials of an authenticatable entity; for example, where a queue manager is an authenticatable entity, queues belonging to that queue manager may share its credentials.

⁸ This topic is discussed in detail in the MQSeries Everyplace Configuration Guide.

⁹ Authenticatable support is more general than this; the certificate server issues certificates to private registries and anything can own such a registry. For example, an application could have its own private registry for use with message-level security.

The name of an authenticable *queue manager entity* is taken as the name of that queue manager; the name of an *authenticatable queue entity* is taken as the string:

`<queue manager>+<queue>`

where: `<queue manager>` is the name of the queue manager and `<queue>` is the name of the queue.

Information on authenticatable entities is stored in the MCS *registry*. The following data is supported:

Address (held as four UNICODE strings of arbitrary content).

Certificates (copies of all issued certificates).

Certificate request authorization details.

The certificate request authorization details control the MCS response to a request for a certificate from an authenticatable entity. The authorization details available are:

Request PIN (the password to be supplied by the requestor with the request).

Duration (the period of time for which the issued certificate will be valid).

Attempts (the number of attempts allowed to request a certificate, after which the authorization to issue will be revoked).

The request PIN is a password, chosen by the MCS operator, which must be supplied by the requesting entity with the request, along with its entity name. It is used to validate that the request is from an authorized source; MCS has no other way of knowing that a request received over the network is actually from the intended authenticatable entity. Passwords are subject to validation processing to eliminate short, repetitive and obvious passwords.

Certificates are issued for periods that are integer multiples of a month, up to a maximum duration of 15 months. The lifetime of a certificate starts from when it is issued, not from when its issuance is authorized.

The *attempts* parameter is used to control repeated attempts to obtain a certificate with a bogus PIN; every unsuccessful attempt decrements the remaining attempts value; when no authorized attempts remain, the certificate request authorization is deleted.

3.3 *Mini-certificates*

Copies of all WTLS issued certificates are stored in the MCS registry. A certificate has the following properties available for inspection:

Subject:	The name of authenticatable entity, followed by attributes.
Issuer:	The name of the issuer, followed by attributes. The issuer name is always of the form MQeMCS@<server>, where <server> is the host name of the MCS server on the network.
Not before date:	The date before which the certificate is invalid.
Not after date:	The date after which the certificate is invalid.
Version:	The WTLS certificate version number.
Algorithm:	The signature algorithm identifier.
Parameter specifier:	Algorithm parameter specifications.
Public key type:	The public key type.
RSA public key exponent:	The exponent element of the public key.
RSA public key modulus:	The modulus element of the public key.
Signature:	The digital signature.

A certificate can only be renewed if a copy is available in the registry; if not a new certificate must be requested. This means that only an MCS server with access to the registry that first issued the certificate can renew the certificate. Certificates can be renewed at any time after issue; they do not need to run to expiry.

All requests for certificate issue or renewal, are typically accompanied by a request for a copy of the MCS's own certificate; this will always be returned with a validity period of at least two years.

3.4 *Registry*

MCS stores certificates and associated data in a secure registry. Multiple registries can exist on a single machine, with the registry to be used being indicated in the profile (through the *registry path* parameter). Each registry represents a quite distinct certificate server, with no sharing of authenticatable entity data or certificates between them. For example, a certificate request authorization held in one registry, is not valid if the certificate is requested from a different registry. Likewise, certificates issued from one registry cannot be renewed from another.

Typically multiple registries are used to distinguish test systems from production systems. A user may be authorized to get certificates from a test server registry; this will not enable that user to request certificates from the production system. In the same way, a certificate obtained by a user from a test server will not be valid for authentication purposes in place of a certificate issued by the production certificate server.

Certificate server administrators use profiles to distinguish between the various certificate servers; users distinguish between them by the IP address, port and protocol that they use to connect.

3.5 Profiles

MCS uses profiles to hold server configuration data. Profile contain the following properties:

- Name: The profile name, which must be unique to each profile.
- Passphrase: The passphrase used to protect the MCS registry. Passphrases must follow the rules given in *Password and passphrase rules* on page 82.
- IP port: The IP port used for incoming connection requests. The port number must match that configured in the remote queue manager.
- IP adapter: The class name of the MQe communications adapter used for the client/server channel from the requesting authenticatable entity. The default value is *com.ibm.mqe.adapters.MQeTcpipHttpAdapter*. This IP adapter must match that configured in the remote queue manager.
- Channel timeout: The time in seconds after which unsatisfied incoming connection requests will time out.
- Max. channels: The maximum number of concurrent channels that are allowed.
- Registry adapter: The class name of the MQe storage adapter used to access the registry store. The default value is *com.ibm.mqe.adapters.MQeDiskFieldsAdapter*.
- Registry path: The registry file path to be passed to the registry adapter. MCS will construct a directory hierarchy below the supplied file path.
- Audit log file: The audit log file to be used for logging MCS activity.
- Start logging when profile loaded: A flag indicating whether audit logging is to automatically started after the profile is opened.

Multiple profiles can exist and can relate to the same registry if desired; however, for a particular MCS registry, only one profile can be open at any one time. Typically profiles are used to distinguish between different registries (through the *registry path* property); multiple profiles to the same registry can be used to change the port and protocol, or the logging options, for that registry. Any one registry requires the same *passphrase* to be used, irrespective of profile.

On Windows platforms MCS stores profiles in the Windows registry; on all other platforms, profiles are stored in the file system.

3.6 *Operator interfaces to MCS*

MCS can be controlled either from the command line or through a Windows graphical user interface. A single MCS registry can be accessed sequentially through either interface; the only restriction being that one MCS registry can only be activated through a single MCS instance at any one time. In this way multiple MCS servers can co-exist on a single machine, provided that they have their own registries and avoid IP port conflicts; in practice however this an unlikely deployment.

Profiles are likewise user-interface independent.

An MCS server, once loaded through the opening of a profile, is always able to accept operator commands, e.g. to display contents or to create or update certificate authorization requests. A loaded MCS server can also be selectively *started* and *stopped*. Starting the server causes it to listening for incoming connection (and hence certificate) requests; stopping it, causes it to ignore connection requests. The server is always loaded in the stopped state.

MCS server activity can be logged to an audit log file; this file contains details of significant operator actions and all network request activity. The log file is appended with activity in each server session and therefore if logging is enabled it is necessary to manage the log file since it will grow in size. In the graphical user interface version of MCS, recent log activity can also be viewed in a report display.

4 Getting started

Two independent examples of getting started follow. If you intend to use the graphical user interface then skip past the sections entitled "Using CommandConsole". If you intend to use the command line interface, read both sections.

The script below demonstrates the use of MCS, including authorizing the issue of credentials and the subsequent request for those credentials by a queue manager. MQe_Explorer is used to create the queue manager and to subsequently inspect its properties. If you have not already installed MQe_Explorer then you should do so now.

The scenario has the following steps:

Using MCS:

- Bring up the certificate server.
- Create (and open) a new profile.
- Authorize the issue of credentials to a queue manager.

Using MQe_Explorer:

- Create a queue manager with a secure registry and certificate credentials.
- Auto-registration with the certificate server (automatic).
- Certificate inspection.

Using MCS:

- Log and summary inspection.
- Viewing the properties of the authenticatable entity.


Using MCS:

- Delete entities created in these examples.

4.1 Loading a profile and creating entities

Using MQe_MiniCertServer

Loading the certificate server

To start the certificate server double click the  icon on the desktop or run the *MQe_MiniCertServer.exe* from a command line. The following window appears:

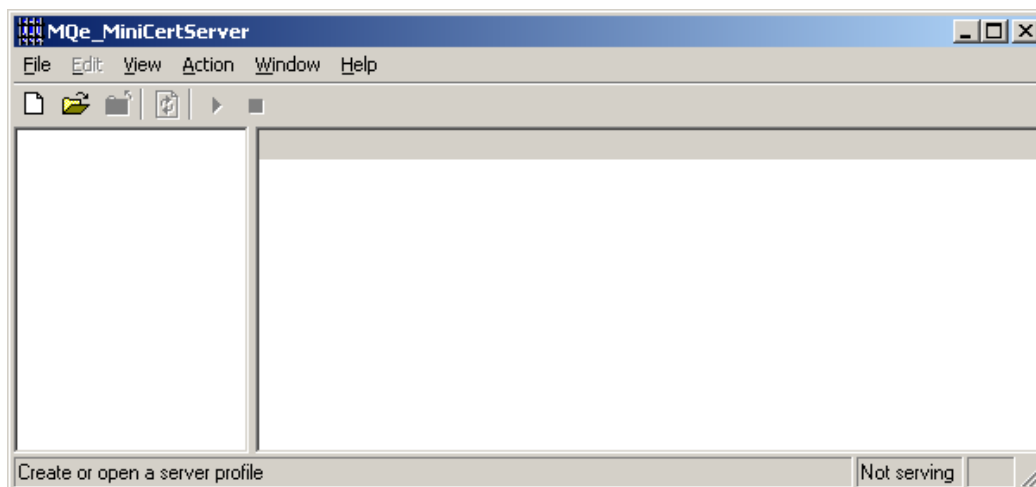


Figure 4-1: The initial MQe_MiniCertServer window

Click *View*→*Report* to activate and display the report window; this will display all recent server and significant operator activity.

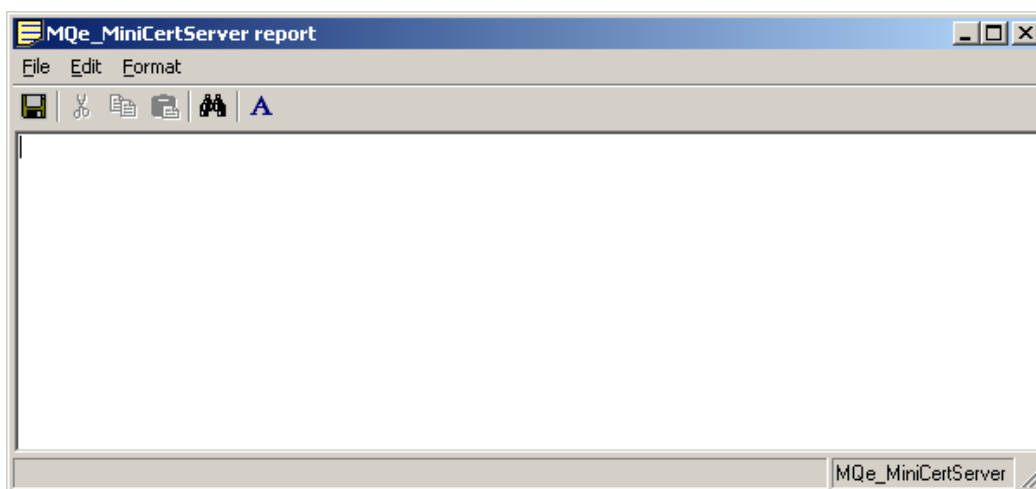



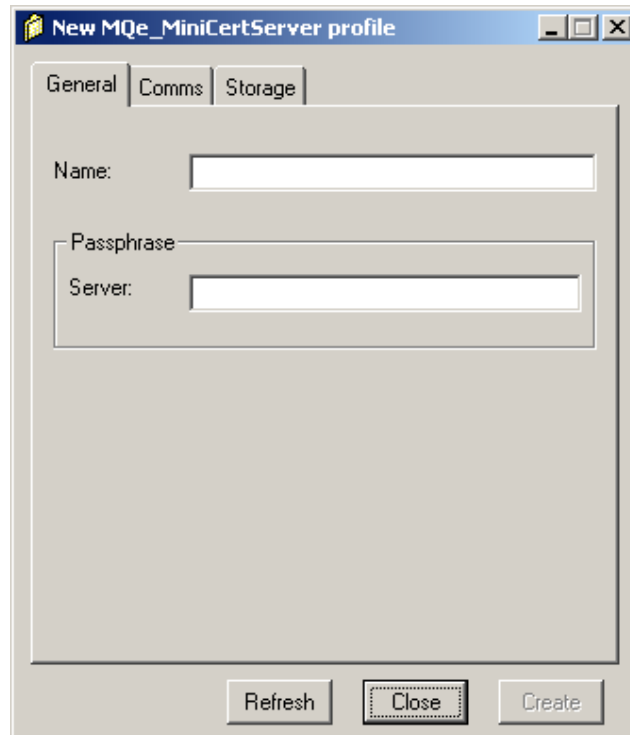
Figure 4-2: MQe_MiniCertServer report window

Activity can also be logged to an audit file; however in this example we will only display events.

Manipulating profiles

Since this is the first time that MCS has been run, a new profile must be created; next time the MQe_MiniCertServer is loaded, an existing profile could be opened instead.

To create a new profile, click the new profile icon  on the toolbar. The form below will be displayed:



The screenshot shows a dialog box titled "New MQe_MiniCertServer profile" with three tabs: "General", "Comms", and "Storage". The "General" tab is selected. It contains two text input fields: "Name:" and "Server:". The "Server:" field is nested under a "Passphrase" label. At the bottom of the dialog are three buttons: "Refresh", "Close", and "Create".

Figure 4-3: Create new profile – General tab

Enter the following data on the *General* tab:

1. **Name:** a unique name for this profile (e.g. "Default").
2. **Passphrase:** a password string to protect the registry (e.g. "abcdefghijkl").

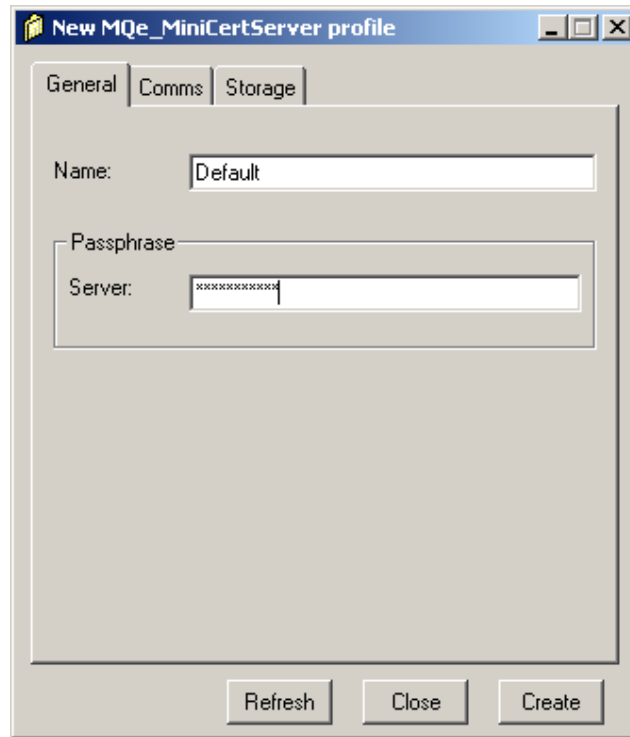


Figure 4-4: Create new profile – General tab, with data entered

Enter the following data on the *Comms* tab:

1. **Port:** the IP port to receive incoming connection requests (e.g. "8085" is the expected default value).
2. **Adapter:** the adapter to be used (by default the value "com.ibm.mqe.adapters.MQeTcipHttpAdapter" is used – this adapter supports incoming requests over the HTTP protocol).
3. **Timeout:** the channel time out value in seconds (by default "300" is proposed).
4. **Max. channels:** the maximum number of simultaneous channels (by default no maximum limit is set).

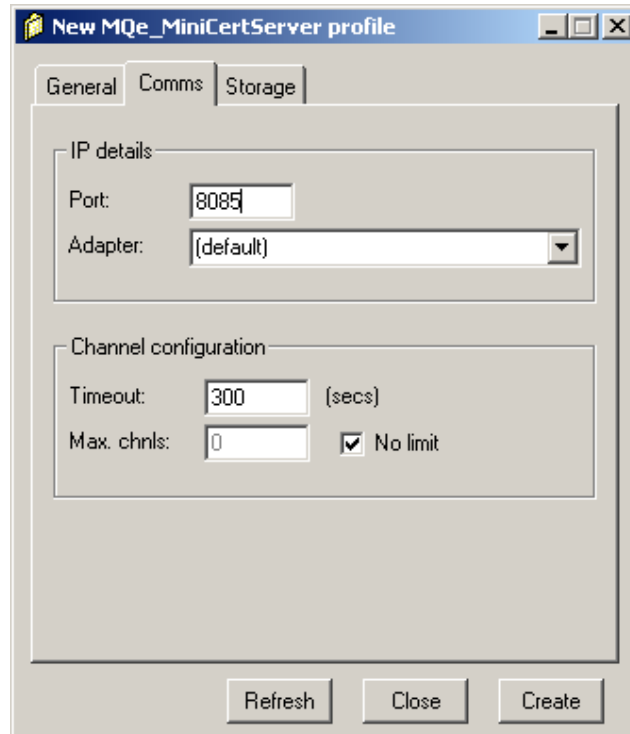


Figure 4-5: Create new profile – Comms tab

Enter the following data on the *Storage* tab:

1. **Adapter:** the storage adapter used to hold the server data (by default the value "com.ibm.mqe.MQeDiskFieldsAdapter" is used).
2. **Path:** the location of the directories and files used to hold the server data.
3. **Log file:** the fully qualified path of a file to be used to hold the audit log data (the file and associated directories will be created if they do not exist).

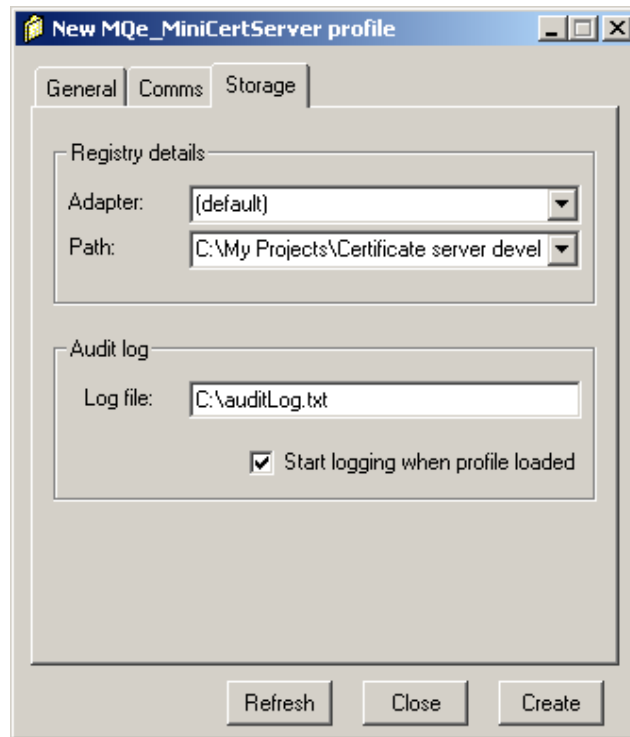


Figure 4-6: Create new profile – Storage tab

Click the *Create* button to save and load the profile. The passphrase will be requested again; after re-entering, the main window will then change in appearance:

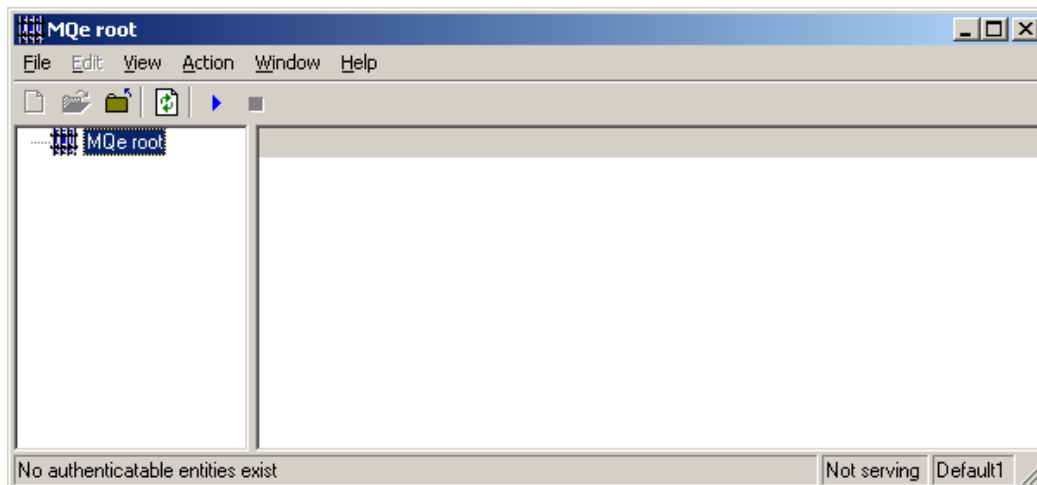


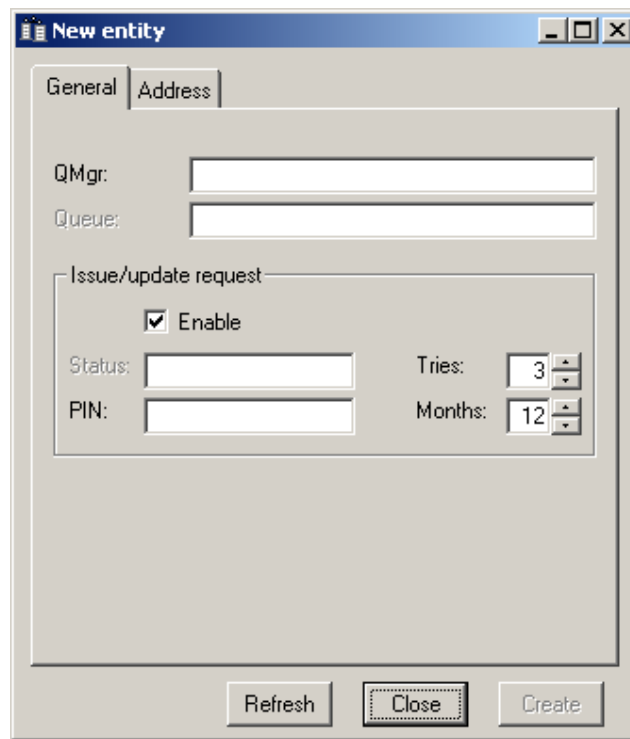
Figure 4-7: The running MQe_MiniCertServer window

The left hand pane holds the tree view of the authenticatable entities; since none exist only the root representing the MCS server (and associated profile) is shown. The right hand pane shows the children of the selected tree node in the left hand pane; again, for the same reason, the pane is empty.

The window display and menus of MQe_MiniCertServer are similar in many ways to those of MQe_Explorer. If you have used the MQe management tool, then you will find MQe_MiniCertServer parallels its operations as far as possible, given their different roles.

Authorizing the issue of a queue manager certificate

To create a new queue manager authenticable entity, right click on the *MQe root* node in the tree pane and select *New Entity* (or select the *MQe root* node and use the *File→New→Entity* menu item). The new entity dialog appears:



The image shows a Windows-style dialog box titled "New entity". It has two tabs: "General" (selected) and "Address". The "General" tab contains the following fields and controls:

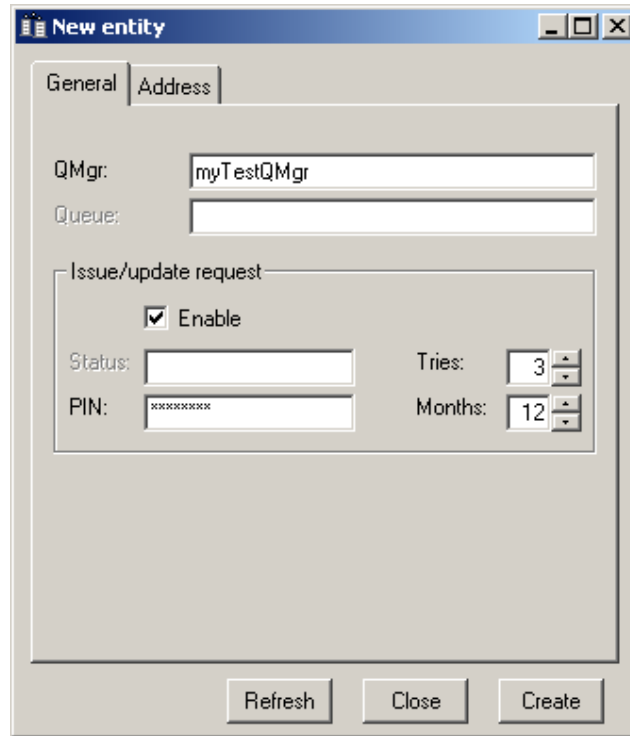
- QMGr:** A text input field.
- Queue:** A text input field.
- Issue/update request:** A section containing:
 - Enable**
 - Status:** A text input field.
 - Pin:** A text input field.
 - Tries:** A spin box set to 3.
 - Months:** A spin box set to 12.

At the bottom of the dialog are three buttons: "Refresh", "Close", and "Create".

Figure 4-8: The new queue manager entity dialog – General tab

Enter the following data on the *General* tab:

1. QMgr: the name of the queue manager (in this example "myTestQMgr")
2. PIN: the mini-certificate request pin to be used by the queue manager when retrieving its certificate (e.g. "12345678")



The screenshot shows a 'New entity' dialog box with two tabs: 'General' and 'Address'. The 'General' tab is active. It contains the following fields and controls:

- QMgr:** A text field containing 'myTestQMgr'.
- Queue:** An empty text field.
- Issue/update request:** A section with a checked 'Enable' checkbox.
- Status:** An empty text field.
- Tries:** A spinner control set to '3'.
- PIN:** A text field containing 'XXXXXXXX'.
- Months:** A spinner control set to '12'.

At the bottom of the dialog are three buttons: 'Refresh', 'Close', and 'Create'.

Figure 4-9: The new entity dialog – General tab, with data entered

The *Tries* and *Months* parameters have been left to their default values. *Tries* controls the number of attempts that are allowed for the authenticatable entity to retrieve its certificate; each time an attempt is made with the wrong PIN, one less attempt remains. *Months* controls the duration of the certificate; by default certificates are issued for 12 months, but the period can be set in the range 1 – 15 months.

Other identification data can be entered on the *Address* tab; each line can contain arbitrary text. MCS stores and retrieves this information, but does not use it.

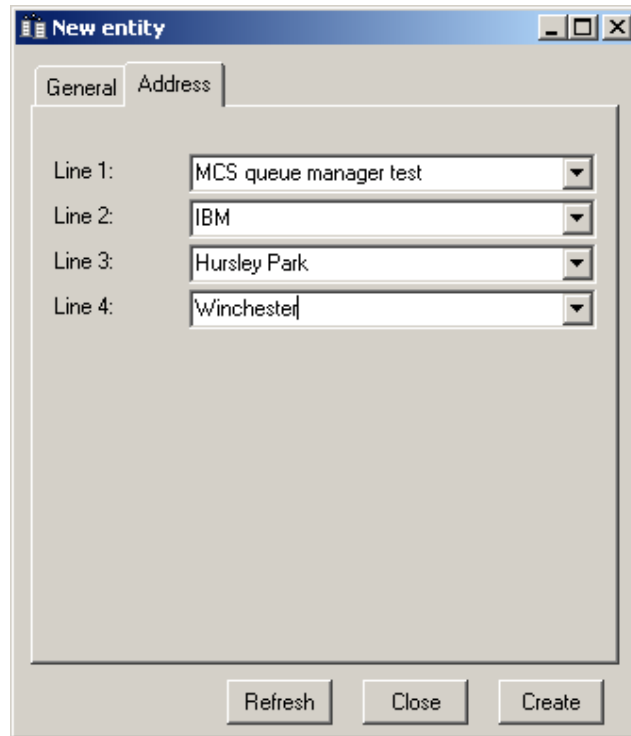


Figure 4-10: The new entity dialog – Address tab

Click on the *Create* button, then the *Close* button to remove the dialog. The tree and list view panes of the MQe_MiniCertServer will be updated; likewise the report window will have logged the activity.

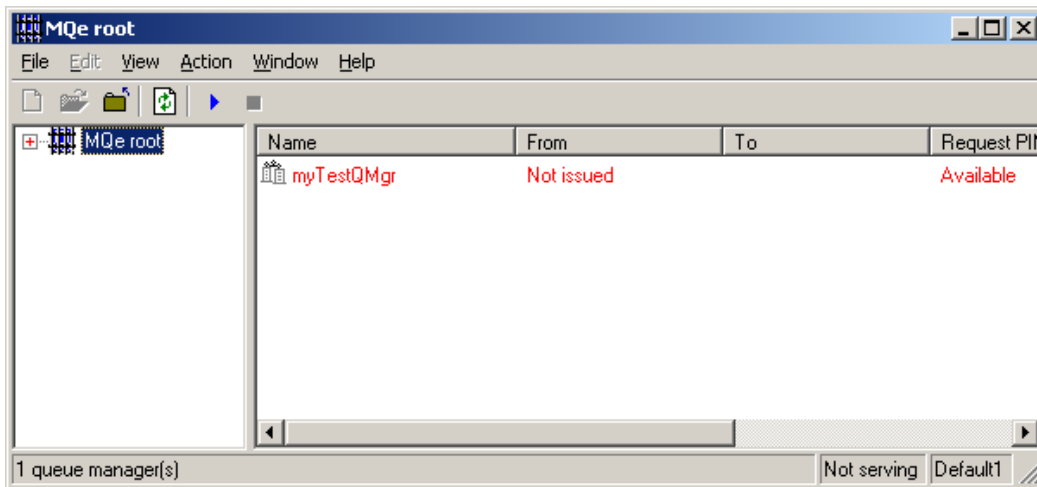


Figure 4-11: Main window before auto-registration

The right hand pane can be scrolled in order to see all the contents. Dragging the dividers between the list view headings will change the column widths. The columns can be re-ordered by dragging the column headers into their new position. In both the list view and tree view panes all the information is displayed in red; this is a visual indication that the server is stopped, i.e. it is not serving certificates in response to network requests.

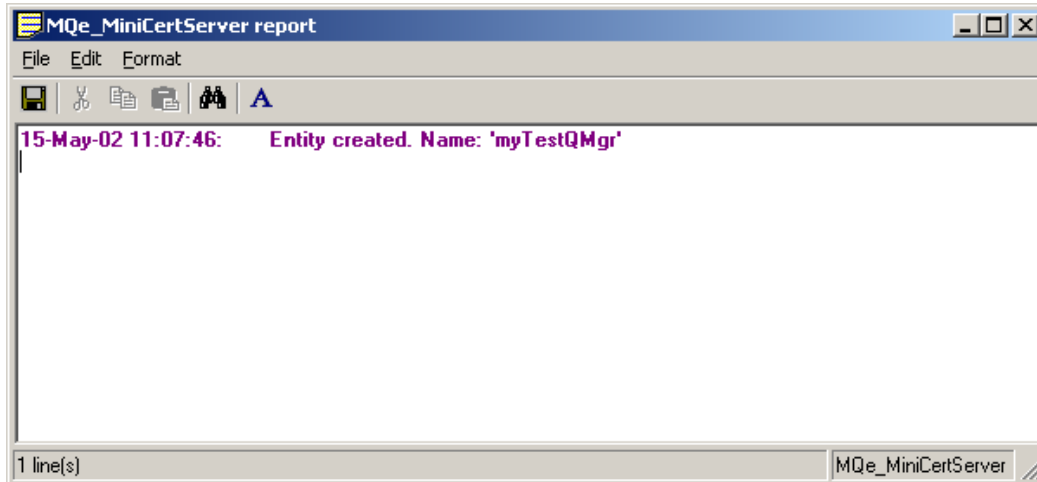



Figure 4-12: Report window before auto-registration

The report window logs the creation of a new authenticatable entity.

Start the MCS server by clicking the  icon on the toolbar (or by using the *Action*→*Start* menu item). The text in red changes to black and the status bar indicates that the server is now listening for incoming certificate requests. In this running state all administrative commands are still fully enabled.

Leave the MQe_MiniCertServer running so that it can serve certificates on demand to requests received over the network.

Using CommandConsole

Loading CommandConsole

To invoke the text command interface program the command line, type

```
java com.ibm.mqe.mqe_minicertserver.CommandConsole
```

The “java” command is the one used for IBM java virtual machine implementations(JVM). jview is the name of the Microsoft virtual JVM, and jre is the name of the Sun JVM. Substitution for the correct JVM implementation name may be required above.

This will produce output similar to:

```
MQSeries Everyplace - Mini Certificate Issuance Server - Command Console
Version 1.0.0.0
Copyright IBM Corp. 2002. All Rights Reserved
US Government Users Restricted Rights - use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Try typing 'mqe_cert_help' for information on valid commands.

>
```

The “>” character indicates that you should type in another command.

Manipulating profiles

On non-Windows systems, MCS stores each profile in a separate file on the file system; in these situations we recommend that all profiles be held in the same directory, making it easy to list the profiles already available, by listing the files in that directory. On Windows systems the profiles are held in the Windows registry.

In this example, we use a default profile *default.profile* that is created and then expected in the current directory.

To create this default profile file in the current directory and view it, use the commands:

```
>mqe_cert_profile -create -passphrase {0 my 1 first 2 secure 3 passphrase 4}  
-port 8085 -timeout 300 -registrydir {C:/Temp} -maxchannels -1  
1  
>mqe_cert_profile -view -verbose 1  
-name default.profile -maxchannels -1  
-commsadapter com.ibm.mqe.adapters.MQeTcipHttpAdapter -port 8085  
-registrydir {C:\Temp} -storageadapter com.ibm.mqe.adapters.MQeDiskFieldsAdapter  
-timeout 300 -passphrasedigest DE233D72676CC4A7E1C2D6182E87E6D282213FBE  
>
```

To load the default profile, use:

```
>mqe_cert_server -load -passphrase {0 my 1 first 2 secure 3 passphrase 4}  
1  
>
```

To start the audit log running with information being stored in the default audit log file *MiniCertServerAuditLog.txt*:

```
>mqe_cert_auditlog -start  
1  
>
```

To start the server issuing certificates in response to network requests:

```
>mqe_cert_server -start  
1  
>
```

Authorize the issue of a queue manager certificate


To authorize a queue manager called "myTestQMgr" to be granted a certificate, if it passes the request PIN of "12345678" in its request over the network, type:

```
>mqe_cert_entity -create -name myTestQMgr -reqpin 12345678 -tries 3 -months 12 -address0  
{MCS queue manager test} -address1 {IBM} -address2 {Hursley Park} -address3 {Winch  
ester}  
1  
>mqe_cert_entity -list  
myTestQMgr  
>
```


Now view the contents of the audit log file *MiniCertServerAuditLog.txt*:

```
...
18-Mar-02 14:15:07      :Set request pin. Requesting entity
name:'myTestQMgr'Pin:'12345678' Time:'8640' Tries:'3'
```

4.2 Creating a secure queue manager with mini-certificates

Use MQE_Explorer to create a client queue manager with a private registry containing its own mini-certificate credentials. After MQE_Explorer is loaded, using the icon  on the toolbar (or the *File*→*New*→*Queue Manager* menu item), bring up the new queue manager dialog.

On the *General* tab complete the input fields as shown:

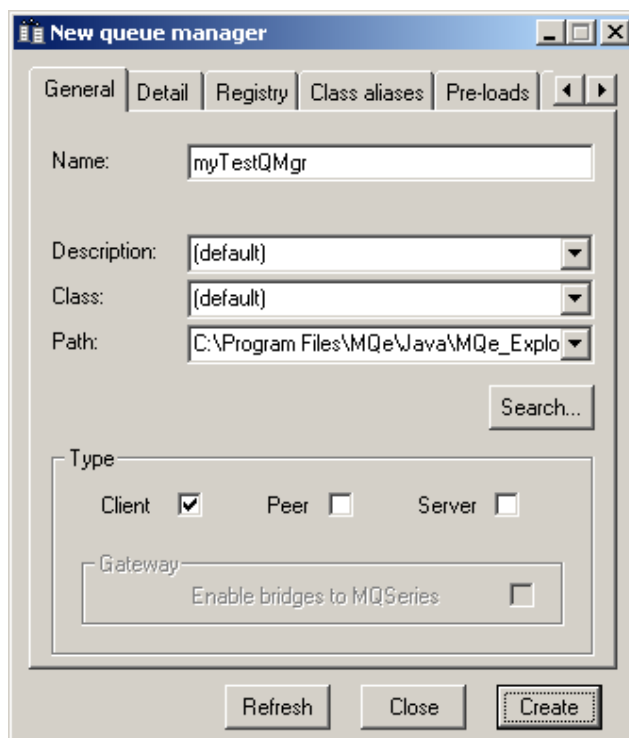


Figure 4-13: Create queue manager – General tab

This *General* tab is used to enter the queue manager name and to set the type to client.

Using the *Registry* tab, set the registry type to *Private registry*:

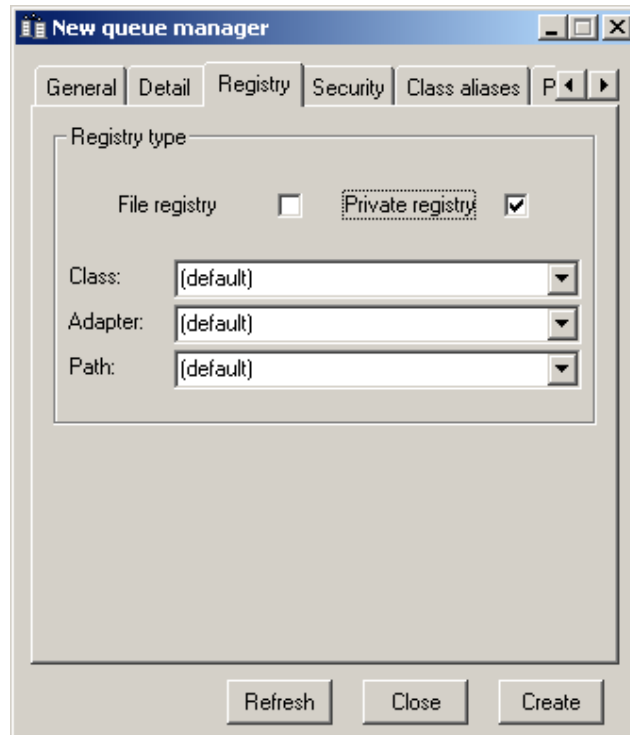


Figure 4-14: Create queue manager – Registry tab

Switch to the *Security* tab.

Use the *Prompt for passwords* option so that passwords are directly requested. Check the *Certificate-based* box to require the queue manager to have certificate-based credentials.

The *IP address*, *IP port* and *Adapter* provide addressability to the mini-certificate server; these values must match those entered previously in the MCS profile. Check the *Allow queue registry* option – this will allow queues on this queue manager to (optionally) have their own credentials. Normally the recommended default is not to allow queue-based credentials.

The default adapter for MQE_Explorer in this case (as for MCS) maps to the value "com.ibm.mqe.adapters.MQeTcpiHttpAdapter".

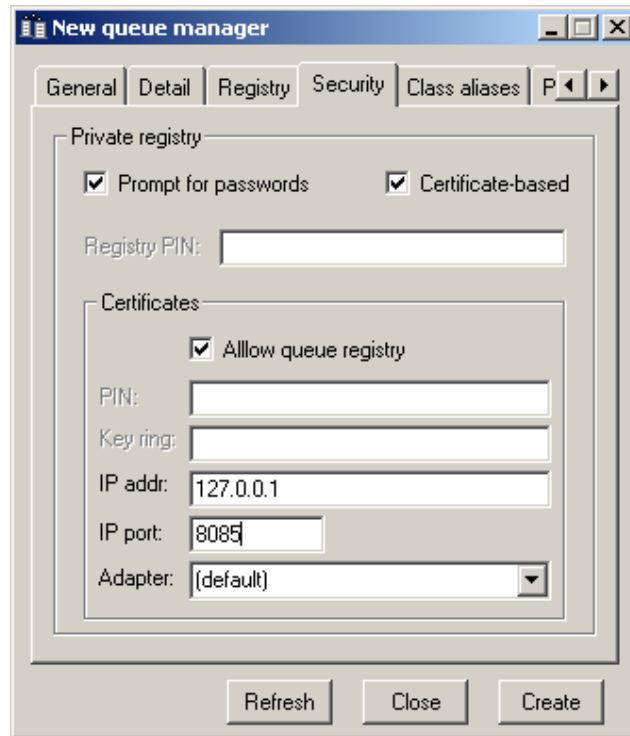


Figure 4-15: Create queue manager – Security tab

After the *Create* button is clicked, MQE_Explorer will prompt for the passwords needed. The first two passwords, registry PIN and key ring password, are private to *myTestQMgr*. The third password is the mini-certificate request PIN, as previously supplied to the MQe_MiniCertServer (e.g. the string: "12345678").

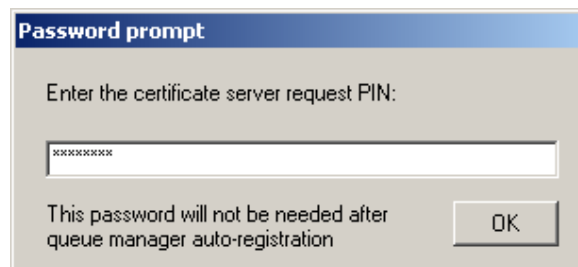


Figure 4-16: Queue manager certificate request PIN prompt

Clicking the *Create* button creates the new queue manager.

During the creation it will obtain the necessary certificates from the mini-certificate server and the details of this auto-registration will appear in the MQe_MiniCertServer's report window:

Certificate inspection

Using MQe_Explorer, details of the certificates acquired are shown in the *Certificates* tab, when the properties of the queue manager are displayed. Thus, selecting the *myTestQMgr* icon in either the list or tree view pane, right clicking and selecting *Properties*, displays the various tabs. The *Certificates* tab has the contents shown below:

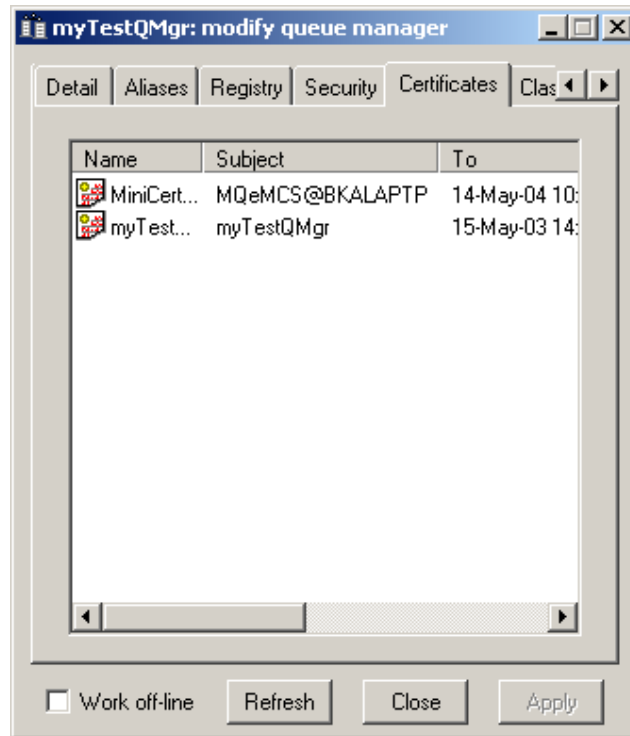


Figure 4-17: Queue manager certificate details

The panel shows two certificates with the name, owner, issuer and validity dates for each; the first relates to MCS, the second to the queue manager itself.

4.3 Log and summary inspection

Using the MQe_MiniCertServer

Viewing activity

Using MQe_MiniCertServer, the report window has been updated to reflect recent activity:

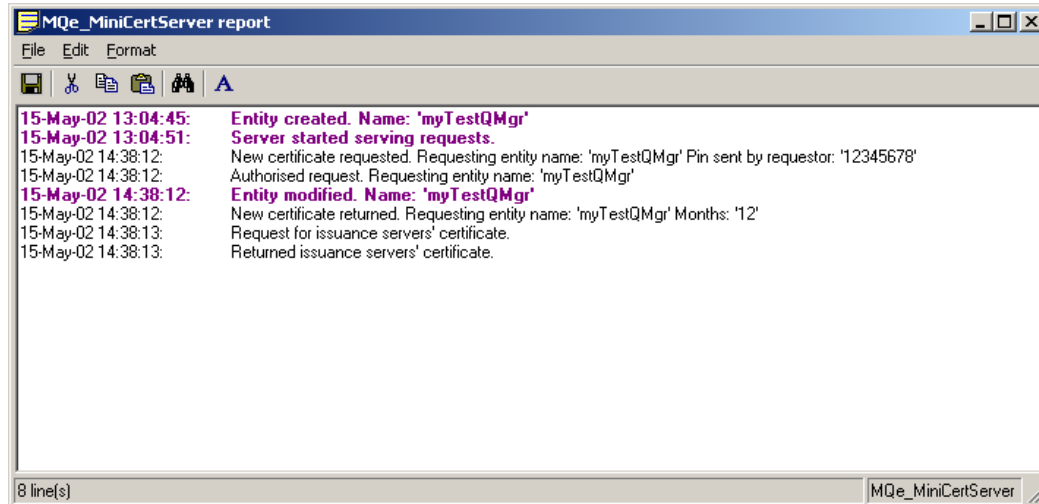



Figure 4-18: Report window after auto-registration

This data is also available in the audit log file.

Two certificates have been issued to the queue manager, its own and that of the certificate authority itself. The PIN used is shown – but the PIN is now invalid and cannot be re-used.

To reflect this network-initiated activity, the summary in the MQe_MiniCertServer list view pane will need to be manually refreshed using the  icon (or *View→Refresh*). The status of the *myTestQMgr* entity has been updated:

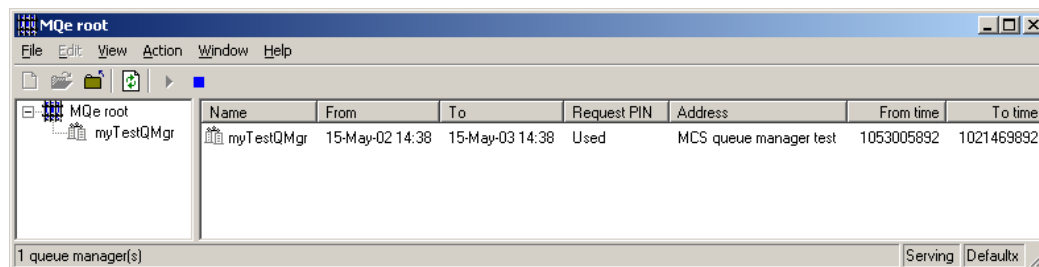
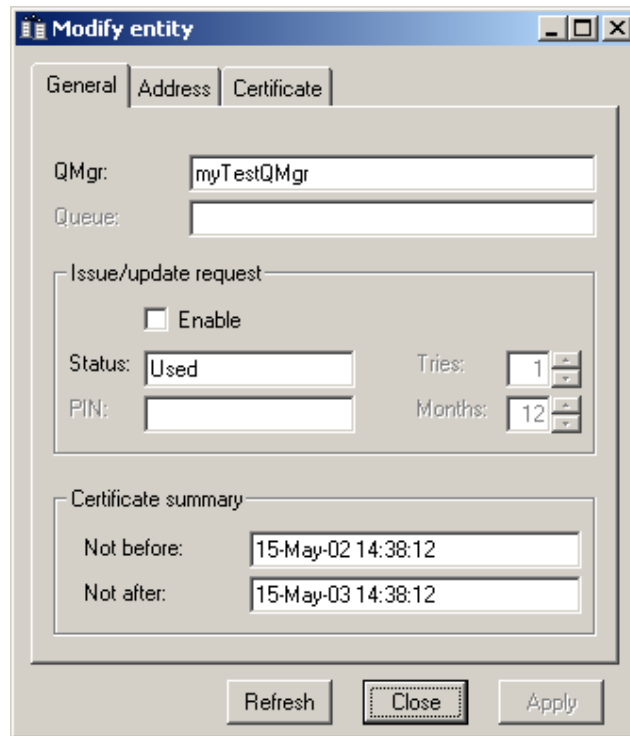


Figure 4-19: The MQe_MiniCertServer window after auto-registration

In this summary, only the first line of the address is shown.

Viewing entity properties

Full details of the authenticatable entity are available through the property pages. Exactly as for MQe_Explorer, right click on the relevant node and select the *Properties* menu item (or use the main menus). The following pages are available:



The screenshot shows a window titled "Modify entity" with three tabs: "General", "Address", and "Certificate". The "General" tab is selected. It contains the following fields and controls:

- QMgr:** Text box containing "myTestQMgr".
- Queue:** Empty text box.
- Issue/update request:** A section containing:
 - Enable
 - Status:** Text box containing "Used".
 - Tries:** Spin box containing "1".
 - PIN:** Empty text box.
 - Months:** Spin box containing "12".
- Certificate summary:** A section containing:
 - Not before:** Text box containing "15-May-02 14:38:12".
 - Not after:** Text box containing "15-May-03 14:38:12".

At the bottom of the dialog are three buttons: "Refresh", "Close", and "Apply".

Figure 4-20: Modify entity – General tab

The *General* tab gives a summary of the entity credential status. The page can be used to authorize renewal of credentials. Certificates once issued, cannot be withdrawn.

Modify entity

General Address Certificate

Line 1: MCS queue manager test

Line 2: IBM

Line 3: HursleyPark

Line 4: Winchester

Refresh Close Apply

Figure 4-21: Modify entity – Address tab

The *Address* tab displays the current address data. This can be updated at any time.

Modify entity

General Address Certificate

Subject: myTestQMgr; ;

Issuer: MQeMCS@BKALAPTP; ;

Cert version: 1 Algorithm: 2

Pub key type: 2 Parm spec: 0

RSA PK exp: 010001

RSA PK mod: 00F74703834248A3E160EED85428A9F
35525B0E03FB954888CAB2B451DAE5
A27F5043903776CF79D4DC8261893C
E906ABDED5F965EFC7374E48867347

Signature: 0D9CB5472CA9255484CA941BAB84C1
66CB9B5791DBEF35AA8E8A35C3D7A
10516081620D90C8677FEF12507D32E
7F04B49F216EFD6CECF8030E5FEAAD

Refresh Close Apply

Figure 4-22: Modify entity – Certificate tab

The *Certificate* tab displays details from the certificate; this information is read-only and cannot be updated. The contents will change if new credentials are issued to the authenticatable entity.

Using CommandConsole

Viewing activity

In previous examples, we used the `mqe_cert_auditlog -start` command, so that information was gathered and deposited into the `MiniCertServerAuditLog.txt` file.

View the file with a text editor.

```
18-Mar-02 14:15:07      :Set request pin. Requesting entity name:'myTestQMGr'  
Pin:'12345678' Time:'8640' Tries:'3'  
18-Mar-02 14:51:23      :New certificate requested Requesting entity  
name:'myTestQMGr' Pin sent by requestor:'12345678'  
18-Mar-02 14:51:23      :Authorised request Requesting entity name:'myTestQMGr'  
18-Mar-02 14:51:23      :Set request pin. Requesting entity name:'myTestQMGr'  
Pin:uncoded value Time:'0' Tries:'0'  
18-Mar-02 14:51:23      :New certificate returned. Requesting entity  
name:'myTestQMGr' Months:'12'  
18-Mar-02 14:51:23      :Request for issuance servers' certificate.  
18-Mar-02 14:51:23      :Returned issuance servers' certificate.
```

Viewing entity properties

To view the changes to the “myTestQMGr” entity in the issuance server, first in non-verbose mode, then in verbose mode:

```
>mqe_cert_entity -view -name myTestQMGr  
-name myTestQMGr -pinstate used -address0 {MCS queue manager test}  
-address1 IBM -address2 {Hursley Park} -address3 Winchester  
-notbefore {18 March 2002 14:51} -notafter {13 March 2003 14:51}  
>  
>mqe_cert_entity -view -name myTestQMGr -verbose 1  
-name myTestQMGr -pinstate used -address0 {MCS queue manager test}  
-address1 IBM -address2 {Hursley Park} -address3 Winchester  
-notbefore {18 March 2002 14:51} -notafter {13 March 2003 14:51} -certversion 1  
-issuer {MQeMCS@COBBETTM; ; } -subject {myTestQMGr; ; } -algorithm 2 -publickeytype 2  
-paramspecifier 0 -signature  
CBB73E5512C1A6EBEC2A2931DA0C67744F18A898F98F0C6CE389FEDC57F71D419E64C56E8EA4  
6F2D0F17D37EE01174BBD4C5226F7A21862CAC0A55F1A0B7B126EECF69093B31BEFB37BD5A80  
CFC828496478F8D1D8FAE8E58F9796AB964002090436068B34E5A7B172370D9BA81F7273E9BA4  
5C7A713DB892CFC6C6FC7A3B4B  
-rsakeyexponent 010001 -rsakeymodulus  
00CDA4E6622175D6B7B5E43158107E0606FEC3490611763  
4213E3BB7AFB8B48E664A038145DCB6877C18AC254E868C2C57FF934B9220D1F292F582224B49  
A69645ADC790532BD3B0EAE2955D2859EC39A9392A4201C9BCF083D01ED355DAA52234A6C38  
BCF40954BB000565F3010026EDEF0296DC60759F03DBDFB911EA9138B8D  
>
```


4.4 Authorizing the issue of a queue certificate

Using MQE_MiniCertServer

Using the MQE_MiniCertServer, right click on the *myTestQMgr* node and select *New Entity*, in either the tree view or list view pane, to display the new entity dialog, primed for a queue authenticatable entity on that queue manager (or select the node and use the *File*→*New*→*Entity* menu item):

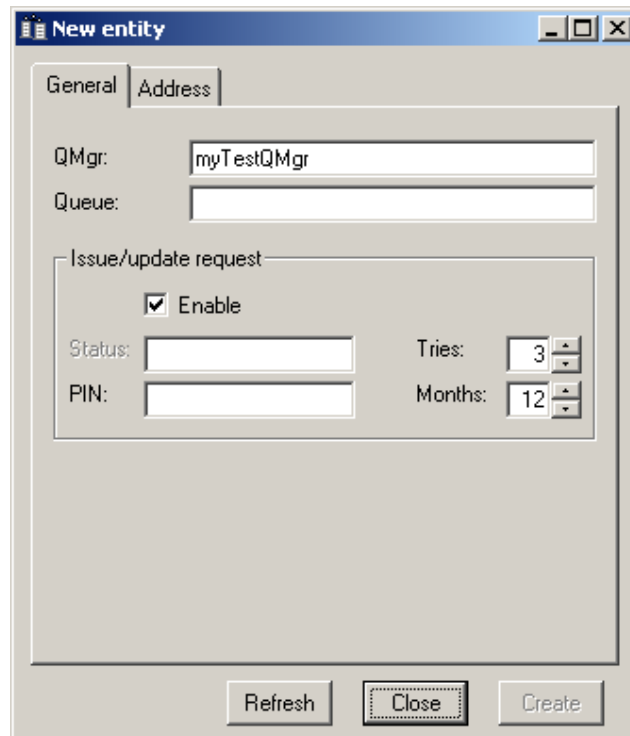


Figure 4-23: The new queue entity dialog – General tab

Type in the name of the queue requiring a certificate. From this point, the remaining steps to authorize the issue of a certificate are identical to those described earlier for a queue manager.

Using CommandConsole

```
>mqe_cert_entity -create -name myTestQMgr+myTestQ -reqpin 12345678 -tries 3 -months 12
1
>
```

4.5 Deleting an authenticatable entity

Using MQE_MiniCertServer

Authenticatable entities can be deleted from the MCS registry by selecting the entity node and invoking the delete command, from the context menu, from the *Edit*→*Delete* menu item, or by hitting the *Delete* key. When an entity is deleted, all information on that entity is lost; this means, amongst other things, that such an entity can no longer have an existing certificate renewed – instead a new one must be issued. Deletion does not withdraw a certificate previously issued.

Using CommandConsole

Delete the queue entity and queue manager entity created previously in the examples above:

```
>mqe_cert_entity -list
myTestQMgr+myTestQ myTestQMgr
>mqe_cert_entity -delete -name myTestQMgr+myTestQ
1
>mqe_cert_entity -delete -name myTestQMgr
1
>mqe_cert_entity -list
>
```

5 MQe_MiniCertServer: graphical user interface

5.1 Main window

Layout

The principal elements of the main window are a menu bar at the top, a toolbar below, two central panes (a tree view pane on the left, a list view pane on the right) and a status bar at the bottom. Individual menu items are described later in this chapter; the various toolbar buttons offer shortcuts to the more commonly used items.

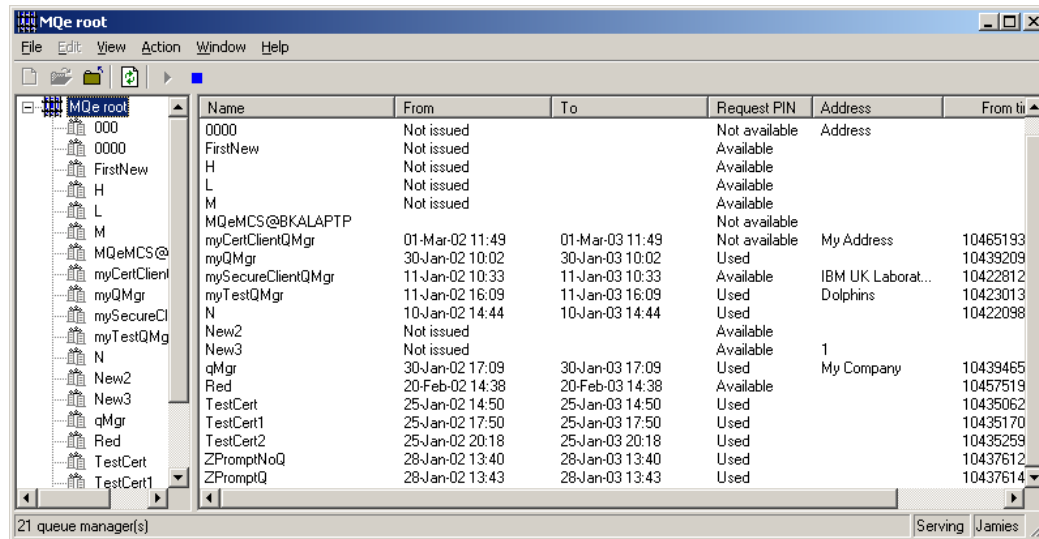





Figure 5-1: An example of the main MQe_MiniCertServer window

The main window can be resized by dragging the sides or the corners; the splitter between the panes can be moved to change the apportionment of real estate between the panes. MQe_MiniCertServer remembers the overall window position and set-up and restores previous settings when re-invoked.

The tree view pane presents the a node tree representing the hierarchy of authenticatable entities; expanding the  symbol next to the MQe root icon , or to a queue manager icon  reveals the next level of entity. Nodes may be selected through a left mouse button click and then actions applied by activating the relevant menu item. A right hand mouse click on a node will bring up a context menu.

The list view pane comprises rows and columns and shows objects or properties that relate to the selected tree node. Single rows may be selected through left mouse button clicks; likewise right hand mouse button clicks are supported. In either case, menu actions can be applied to the selected node – exactly as described for tree view nodes. Columns can be re-ordered by dragging the column header to a new position (that is, over other headers); columns can be re-sized by dragging the column boundaries in the header. Clicking a column header sorts the data by that column; re-clicking re-sorts in the reverse sequence. Double clicking a row in the list view pane is identical to selecting the equivalent object in the tree view pane and will result in a change in the contents of the list view pane.

The status bar contains three panels; these are, from left to right:

- *Status and information message area*
- *Certificate serving status ('Serving' or 'Not serving')*
- *Profile name*

Navigation and control

MQe_MiniCertServer supports the standard Windows conventions for selection, shortcuts, context menus, tabbing through fields and keyboard and clipboard operations.

Multiple object selection is supported (in the appropriate windows) using shift key and control key combinations with the mouse select button. *Control A* selects everything in the list pane (when multiple selections are enabled). Keyboard shortcuts are displayed on the menu items where applicable. Right clicking on an object will reveal context menus for that object – and is supported for all list view and tree view objects. *Tab* will move to the next object in a panel or window – for ease of data entry. *Cut*, *copy* and *paste* functions to/from the clipboard is supported for report data and trace. The complete set of supported keyboard operations is listed in *Accessibility features* on page 80.

Menus

File

New

This command has two sub-commands:

- Profile
 - Entity
 - To create a new queue manager entity, select the MQe root node
 - To create a new queue entity, select the parent queue manager
- Selections are valid in either the tree view or list view panes

Profile

File→*New*→*Profile* displays the new profile dialog used to create a profile. A profile is needed to start the server and to determine its access, network and other operational characteristics.

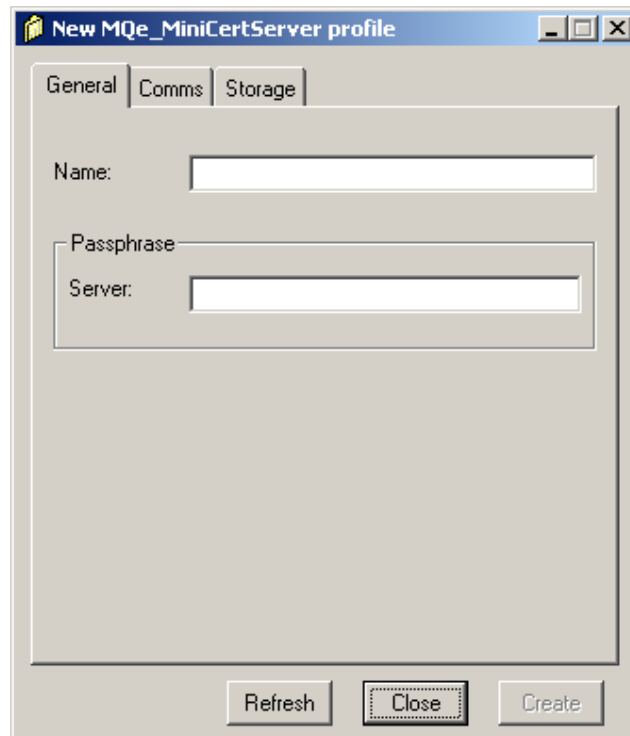


Figure 5-2: New profile dialog – General tab

The dialog has three tabs, each relating to a set of characteristics:

- *General*
The name of the profile and the passphrase required to activate the server. Passphrases, once set, cannot be changed or displayed.
- *Comms*
The network configuration details that determine, amongst other things, how queue managers access the server to obtain and update certificates.
- *Storage*
Registry information determining how and where the server will store certificates and other authenticatable entity data; also information relating to logging of server activity.

The following input fields are present:

- *General* tab (illustrated above):
 - *Name* – the profile name (which must conform to the MQE object naming rules).
 - *Server passphrase* – a character string that controls access to the mini-certificate server registry where the entity and certificate data is stored (this must conform to the MQE_Explorer/MQe_MiniCertServer passphrase rules¹⁰). All profiles targeted at the same registry (identified by the *registry path* property on the *Storage* tab) must use the same value for the *passphrase*.

The *Comms* tab has the appearance:

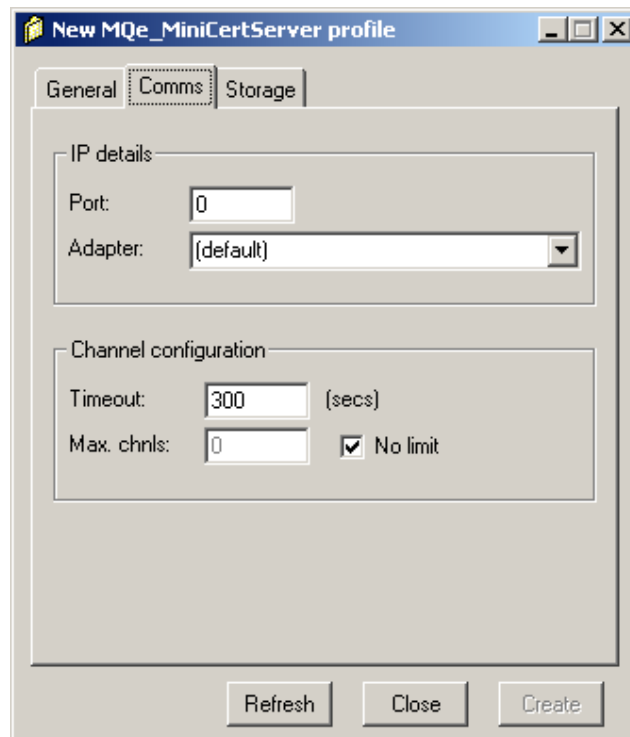


Figure 5-3: New profile dialog – Comms tab

¹⁰ See *Password and passphrase rules* on page 82 for more details.

- **Comms tab:**
 - *IP port* – the port number used by MCS to service incoming connection requests¹¹.
 - *IP adapter* – the class name of the adapter used for network communications¹².
 - (*default*) is mapped to "com.ibm.mqe.adapters.MQeTcpipHttpAdapter".
 - *Timeout* – the timeout in seconds to be used on incoming client/server connection requests.
 - *Max. channels* – the maximum number of concurrent channels that are to be supported at any one time. Checking *No limit* removes any numeric restriction.

The *Storage* tab has the appearance:

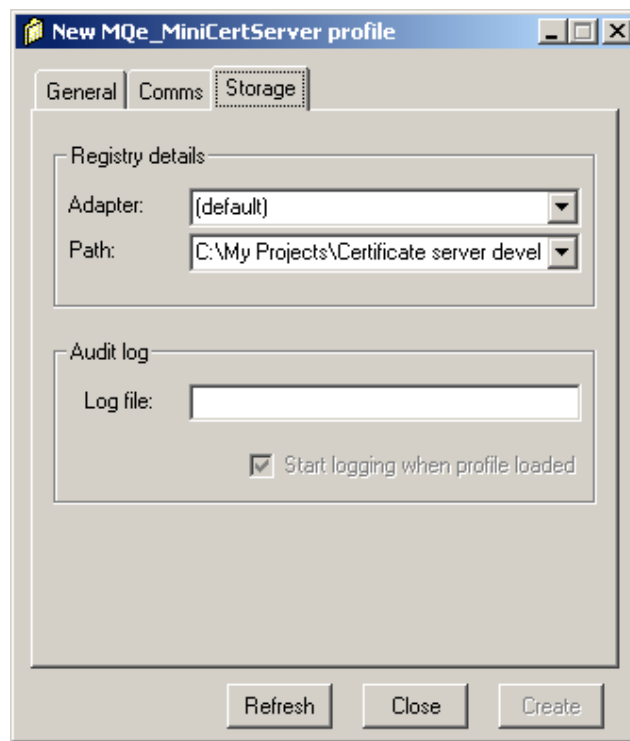


Figure 5-4: New profile dialog – Storage tab

¹¹ It is conventional, but not required, to use the value 8085.

¹² Typically this is set to use the HTTP protocol (by using the adapter *com.ibm.mqe.adapters.MqeTcpipHttpAdapter*) so that requests can easily pass through firewalls. If HTTP is used, ensure that any proxy settings on the system are correctly configured.

- **Storage tab:**
 - *Adapter* – the class name of the adapter for permanent storage access of registry information.
 - *(default)* is mapped to "com.ibm.mqe.adapters.MQeDiskFieldsAdapter".
 - *Path* – the location of the directory below which MCS will construct the registry.
 - *Log file* – the fully qualified name of a file (including the file type extension) that will be used to hold the audit log data. If the file does not exist it will be created; if the file already exists, audit log data will be appended to the existing file contents. The data is in the form of simple text; the file type may be freely chosen, but *.txt* or *.log* would be conventional.
 - *Start logging when profile loaded* – if checked, audit logging is automatically enabled; if unchecked, audit logging may be started through the *File→Audit Log* menu item.

The *Create* button creates the new profile; *Refresh* refreshes the available values in the dialog; *Close* removes the window.

Existing profiles can be edited or deleted by using the *File→Open* menu item and then clicking the *Edit* or *Delete* buttons displayed in the dialog. Alternatively, the current profile can be inspected through the *File→Properties* menu item, when the mini-certificate server icon is selected in the tree.

Entity

File→New→Entity displays the new entity dialog used to create an authenticatable entity.

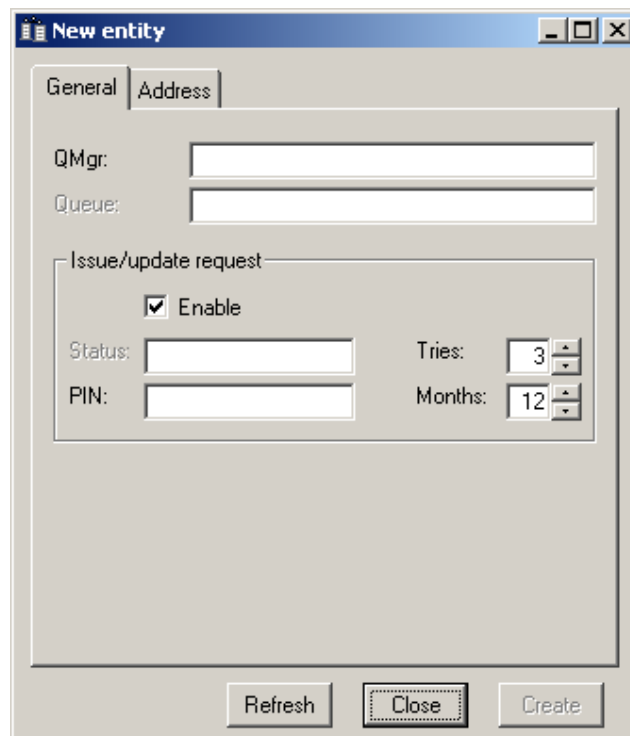


Figure 5-5: New entity dialog for a queue manager – General tab

The dialog has two tabs, each relating to a set of characteristics:

- *General*
The name of the entity and information associated with certificate requests.
- *Address*
Arbitrary strings describing the entity.

The following input fields are present and available:

- *General* tab (illustrated above):
 - *Queue manager* – the corresponding queue manager name.
 - *Queue* – the corresponding queue name (if applicable).
 - *Enable issue/update* – if checked, the issue of a certificate is authorized.
 - *PIN* – the certificate request PIN to be supplied by the authenticatable entity to obtain (or update) its credentials (which must conform to the MQe_Explorer/MQe_MinicertServer password rules).
 - *Tries* – the number of attempts allowed to the authenticatable entity to retrieve its credentials (range 1 – 9).
 - *Months* – the period in months for which the certificate will be valid (range 1 – 15). Certificates are dated from the time that they are issued, not the time from which their issue is authorized.

The *Address* tab has the appearance:

Figure 5-6: New entity dialog – Address tab

- *Address* tab:
 - *Line 1* – an arbitrary string describing the entity (this line is also shown in the summary list view pane).
 - *Line 2* – an arbitrary string describing the entity.
 - *Line 3* – an arbitrary string describing the entity.
 - *Line 4* – an arbitrary string describing the entity.

The *Create* button creates the new profile; *Refresh* refreshes the available values in the dialog; *Close* removes the window.

Open

This command is used to load a profile and launch the mini-certificate server. It also allows profiles to be edited or deleted. The following dialog is displayed:

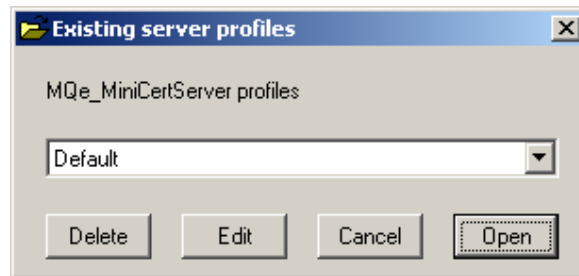


Figure 5-7: Open profile dialog

The available profiles may be selected from the list box. The *Open* button opens a profile and prompts for the associated passphrase, before launching the server. The server is initially placed in the stopped state, i.e. it will not accept network requests for certificates but is available for administration. If subsequently started, it will then accept network requests until stopped; it will still be available for administration.

Delete deletes the selected profile, *Edit* allows the selected profile to be updated, *Cancel* closes the dialog.

When a profile is edited, the profile property pages are displayed, e.g.

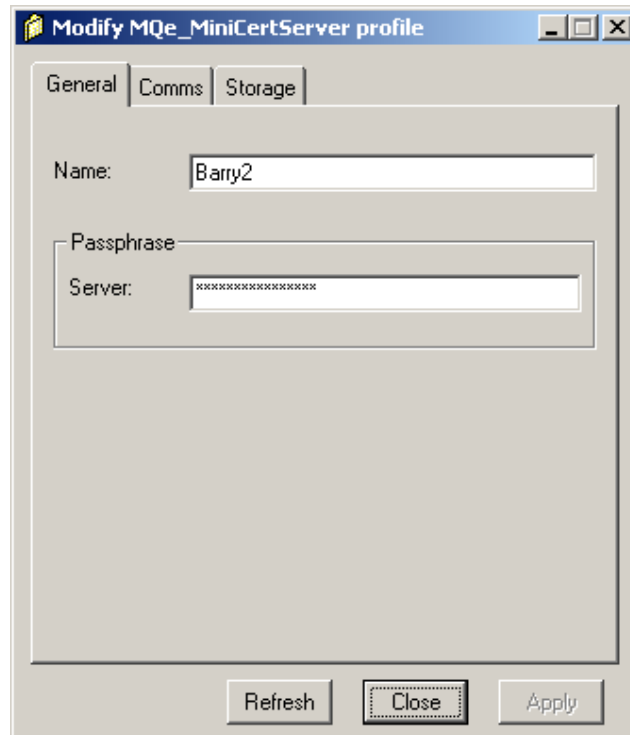


Figure 5-8: Edit profile dialog

These pages are almost identical to those shown when creating a new profile; however only data on the *Comms* tab can be changed. The *Apply* button updates the profile; *Refresh* refreshes the available values in the dialog; *Close* removes the window.

Close

The *Close* command closes the current profile, stops the server if necessary and unloads it. Consequently the server is no longer able to serve certificates and is not available for administration.

Audit Log

The Audit Log command records operator and network request activity in the audit log file associated with the profile. This same information is independently available in the report window (accessed via the *View*→*Report* menu item).

Properties

This command loads the properties pages for the object selected in the tree view or list view panes, i.e. one of:

- *Server*
- *Queue manager entity*
- *Queue entity*

When the server is selected, the current profile is displayed, which may be viewed but not edited. If changes are required, then the current profile should be closed and the profile then edited via the *Edit* button displayed from the *File*→*Open* menu dialog.

A typical entity property page is:

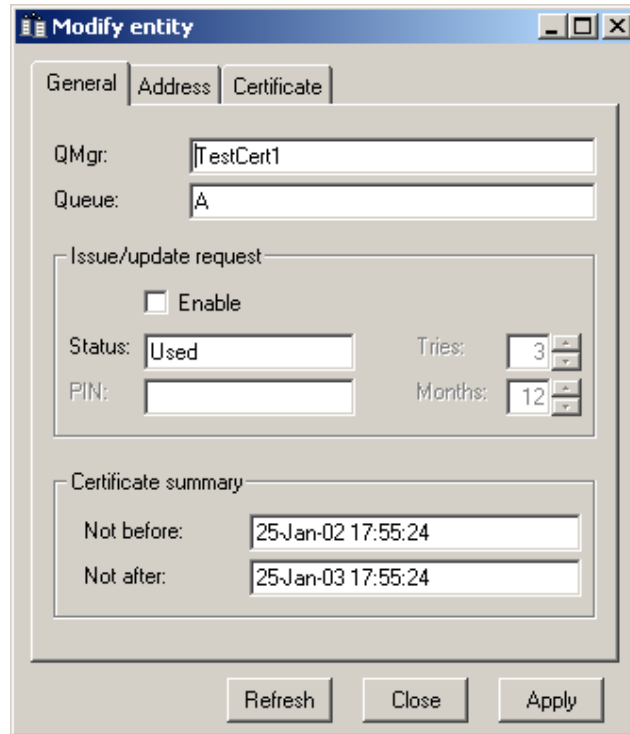


Figure 5-9: Property entity dialog for a queue – General tab

The dialog has a maximum of three tabs, each relating to a set of characteristics:

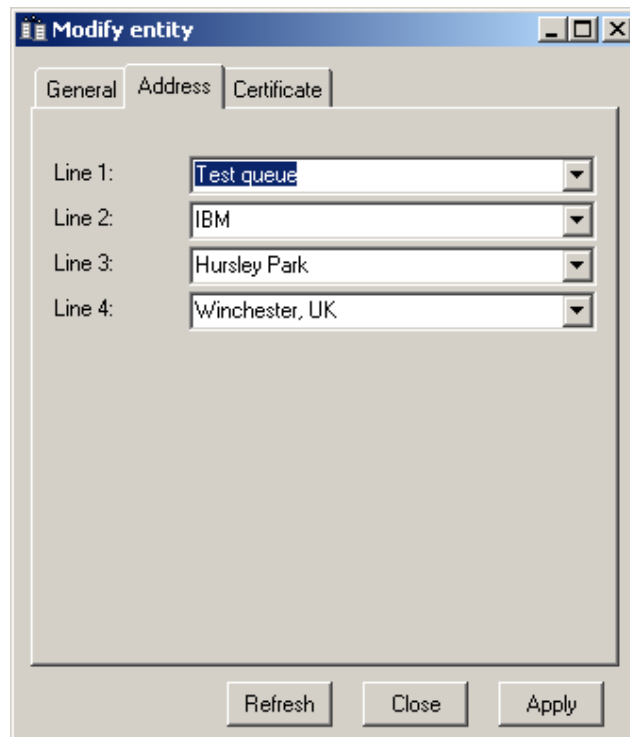
- *General*
The name of the entity and information associated with certificate requests.
- *Address*
Arbitrary strings describing the entity.
- *Certificate*
Details of the last certificate issued.

The following input fields are present:

- *General* tab (illustrated above):
 - *Queue manager* – the corresponding queue manager name.
 - *Queue* – the corresponding queue name (if applicable).
 - *Enable issue/update* – if checked, the issue (or renewal) of a certificate is authorized.
 - *Status* – the last known status (if any) of certificate issuance; one of:
 - *Available* (certificate available).
 - *Not available* (certificate not available).
 - *Used* (certificate has been issued).
 - *PIN* – the certificate request password to be supplied by the authenticatable entity to obtain (or update) its credentials (which must conform to the MQe_Explorer/MQe_MiniCertServer password rules¹³).
 - *Tries* – the number of attempts allowed to the authenticatable entity to retrieve its credentials (range 1 – 9).
 - *Months* – the period in months for which the certificate will be valid (range 1 – 15). Certificates are dated from the time that they are issued, not the time from which their issue is authorized.
 - *Not before* – the date before which the last issued certificate is not valid.
 - *Not after* – the date after which the last issued certificate is not valid.

¹³ See *Password and passphrase rules* on page 82 for more details.

The *Address* tab has the appearance:



The screenshot shows a window titled "Modify entity" with three tabs: "General", "Address", and "Certificate". The "Address" tab is selected. Inside the dialog, there are four text input fields labeled "Line 1:", "Line 2:", "Line 3:", and "Line 4:". The values entered are "Test queue", "IBM", "Hursley Park", and "Winchester, UK" respectively. At the bottom of the dialog, there are three buttons: "Refresh", "Close", and "Apply".

Figure 5-10: Property entity dialog – Address tab

- *Address* tab:
 - *Line 1* – an arbitrary string describing the entity (this line is also shown in the summary list view pane).
 - *Line 2* – an arbitrary string describing the entity.
 - *Line 3* – an arbitrary string describing the entity.
 - *Line 4* – an arbitrary string describing the entity.

The *Certificate* tab has the appearance shown below; none of the fields may be modified.

The screenshot shows a 'Modify entity' dialog box with the 'Certificate' tab selected. The fields are as follows:

- Subject: TestCert1+A; ;
- Issuer: MQeMCS@BKALAPTP; ;
- Cert version: 1
- Algorithm: 2
- Pub key type: 2
- Parm SS: 0
- RSA PK exp: 010001
- RSA PK mod: 00E9EC92D42D33A5CDEF9D811ABB8
F7A5287FE91937AF2041A2A5188DE47
810B00DC917F109A2B7357F49755095
34AB13C481405EB4DE3CDD4E0C7E
- Signature: 209A871B3426DBC99D3383B1BD474B
7560571711E54CA9F7E7A7FD7E2AF9
5461E94CD6AC90699DC0C5081CE484
F099ACC2F2B06ECAFF419E435DEDD

Buttons at the bottom: Refresh, Close, Apply.

Figure 5-11: Property entity dialog – Certificates tab

- *Certificate* tab:
 - *Subject* – the subject name, plus by any attributes.
 - *Issuer* – the issuer name, followed by attributes.
 - *Certificate version* – the certificate version number.
 - *Algorithm* – the signature algorithm identifier.
 - *Parameter specifier* – the algorithm parameter specifier.
 - *Public key exponent* – the exponent element of the public key.
 - *Public key modulus* – the modulus element of the public key.
 - *Signature* – the digital signature.

The *Apply* button modifies the properties; *Refresh* refreshes the available values in the dialog; *Close* removes the window.

Exit

Exit terminates execution of MQe_MiniCertServer.

Edit

Delete

Delete deletes a authenticatable entity, identified as the one selected in the tree view or list view panes. All properties are removed and any copies of previously issued certificates to that entity are destroyed.

View

Report

This menu items displays the Report window that displays audit log information for both operator- and network-initiated actions. For more detail see *Report window* on page 49.

Trace

This menu item displays the *Trace window*, used by IBM service personnel. For more details see *Trace window* on page 51.

Small icons

This view option affects the list view pane – entities are displayed as the entity name together with an associated icon, with multiple entities per row. No other entity properties are available.

List

This view option affects the list view pane – entities are displayed as the entity name together with an associated icon, with a single entity per row. No other entity properties are available.

Details

This view option affects the list view pane – entity properties are displayed as a row of column values. The first column holds the entity name together with an associated icon. The remaining columns display other entity properties.

Refresh

Refresh refreshes the displayed information for the selected object and its children in both the tree and the list view panes. Use of *Refresh* is necessary to see the current state of an object where it may have been externally updated through network requests.

Action

Start

The *Start* menu item enables the mini-certificate server to listen for and process incoming network requests for certificates.

Stop

The *Stop* menu item disables the mini-certificate server from listening for and processing any incoming network requests for certificates.

Window

Close all

Close All closes all open windows, excepting: the main window, the trace window (if present) and the report window (if present).

Minimize all

Minimize All minimizes all open windows, excepting the main window.

Restore all

Restore All restores all minimized windows.

Help

About MQe_MiniCertServer

This menu item lists the program name, product identifier and version. The window is closed via the system close icon in the top right hand corner. The *System Info* button provides information on the local environment – IBM service staff may require this. The information available is as below:



Figure 5-12: System information panel

In the formatting of the *classpath* variable value, a space is added after every semi-colon – this is to enable the string to word spill across multiple rows, if necessary. If any values (such as file paths) are not completely visible, because they are long strings without embedded blanks, then the value may be clicked and the cursor moved to the right – the value will then scroll in the field. All values can be selected and copied to the system clipboard.

List view pane

The list view pane displays the child (or children) of the object currently selected in the tree view pane. If the detailed view mode has been set, then a tabular view that includes detailed characteristics is displayed. To list the queue manager entities, select the mini-certificate server icon in the tree view pane; to list queue entities, select the appropriate queue manager entity in the tree view pane. For convenience, when a queue entity is selected in the tree view, that queue's details are shown in the list view pane.

The list view pane offers a number of functions:

- Rows are sorted (in increasing and decreasing column order) by clicking (or re-clicking) the column header.
- Dragging the column boundaries changes column widths.
- Columns are re-ordered by dragging the column titles.

Only one row may be selected at a time. Drag & drop operations are not supported.

The status bar contains three sub-panels; from left to right:

- Object status and information.
- Certificate service status (*running* or *stopped*).
- Current profile name.

Entities

The following entity information is displayed; note that the full set of properties for an entity is available by selecting the entity in either the tree view of list view panes and using the context *Properties* menu item or the *File*→*Properties* item:

<i>Name:</i>	The name of the authenticatable entity.
<i>From:</i>	The date/time from which its certificate is valid (blank if no certificate has been issued).
<i>To:</i>	The date/time after which its certificate will be invalid (blank if no certificate has been issued).
<i>Request PIN:</i>	The status of the certificate request PIN; one of: <ul style="list-style-type: none"> • <i>Available</i> – a certificate will be issued on receipt of a valid request. • <i>Not available</i> – a certificate will not be issued in response to any request; nor has a certificate been issued since the request PIN property was last reset. • <i>Used</i> – a certificate has been issued since the request PIN was last reset; a further certificate will not be issued in response to any further request.
<i>Address:</i>	The first line of the address property.
<i>From time:</i>	An integer representation of the date/time after which its certificate is valid (this format of the <i>From</i> property allows sorting in date/time sequence).
<i>To time:</i>	An integer representation of the date/time after which its certificate will be invalid (this format of the <i>To</i> property allows sorting in date/time sequence).

Double clicking an entity in the list view pane is equivalent to selecting that same entity in the tree view pane.

5.2 Report window

The *Report* window provides a display of the current audit log activity. If audit logging is enabled (*File*→*Audit Log*) and the report menu item is enabled, then all items being logged are also written into the report window. Items logged before the report window display was enabled are not shown and, if logging is stopped, current window contents remain. If necessary, the report window trims the content such that only last 500 lines remain visible.

The window supports the display of rich text format (*.rtf*) and data from this area may be cut, copied or pasted to/from other applications, for example Microsoft Notepad, Wordpad or Word. Text may also be highlighted, colored, annotated, cleared, searched or saved in a file. MQe_MiniCertServer supports a subset of text editing capabilities, including word select, deletion, backspacing, tab insertions, etc.

An example of the report window is shown below:

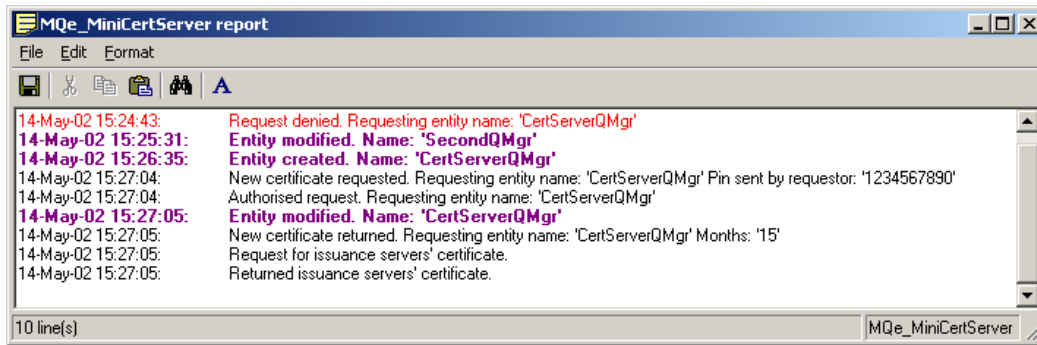


Figure 5-13: The report window

Menus

File

Save as

Save As saves the contents of the report window as either a plain text file (.txt) or as a rich text file (.rtf).

Exit

Exit stops reporting of audit log entries and removes the trace window.

Edit

Cut

Cut cuts the currently selected text to the system clipboard. The text is stored in both rich text and plain text formats.

Copy

Copy copies the currently selected text to the system clipboard. The text is stored in both rich text and plain text formats.

Paste

Paste pastes text from the system clipboard to the window. If text at the destination is selected, it is replaced; otherwise the clipboard contents are inserted at the current text cursor position.

Find

Find locates a text string in the window and, if found, leaves the result selected. If *Find* is invoked with text selected, then that selected string is primed as the search string. All search strings used in an invocation of *Find* are available for re-use in the drop down list box. Repeat identical searches begin where the previous search finished, until all the target text area has been searched. Options are provided for both case matched and whole word searches.

Clear

Clear deletes the content of the window.

Delete

Delete deletes the currently selected text.

Select all

Select All selects all the text in the window.

Format**Font**

Font formats the currently selected text; the font, style, size, color and effects may be set.

5.3 Trace window

The trace window provides a tool for IBM service personnel – it is not intended for normal use. The information below is for reference purposes only.

The trace window traps the two output streams *System.out* and *System.err* and allows them to be re-directed to the display; additionally *System.err* may be logged to a file. The text area used to display output supports the rich text format (.rtf) and data from this area may be cut, copied or pasted to/from other applications, for example Microsoft Notepad, Wordpad or Word. Text may also be highlighted, colored, annotated, cleared, searched etc. MQe_MiniCertServer supports a subset of text editing capabilities, including word select, deletion, backspacing, tab insertions, etc.

The nature of the *System.err* data collected is controlled through the *Action* and *View* menus. The *Action* menu determines the items that are traced – these are documented in the MQe publications, but in summary, MQe trace message types are traced as follows:

- _*: displayed if *System* is checked.
- i*: displayed if *System & Information* are checked.
- w*: displayed if *System & Warning* are checked.
- e*: displayed if *System & Error* are checked.
- s*: displayed if *Security & Error* are checked.
- d*: displayed if *System & Debug* are checked.
- I*: displayed if *Information* is checked.
- W*: displayed if *Warning* is checked.
- E*: displayed if *Error* is checked.
- S*: displayed if *Security* is checked.
- D*: displayed if *Debug* is checked.
- Calls to debug*: displayed if *Calls to debug* is checked.

The *View* menu options *Thread names*, *Timestamp* and *Object names* determine whether those characteristics are included in the traced data.

A continuous trace of the *System.err* data can be spooled to a disk file in plain text format (.txt); this is enabled via the *File*→*Log System.err* menu item. The same data can also be displayed in the text area under the *System.err* tab. However, in the latter case the display can be stopped and started via the *View*→*System.err* menu item – note that stopping and starting has no effect on disk file logging. The contents of the *System.err* text area can be saved, either as an .rtf or .txt file, through the *File*→*Save As* menu item (the *System.err* contents must be uppermost in the window).

In a similar manner *System.out* is shown in the text area under the *System.out* tab; the contents can be saved via the *File*→*Save As* menu item when the *System.out* contents are uppermost. There is no filtering of *System.out* – and no disk logging, however the editing commands are fully supported.

An example of the trace window is shown below:

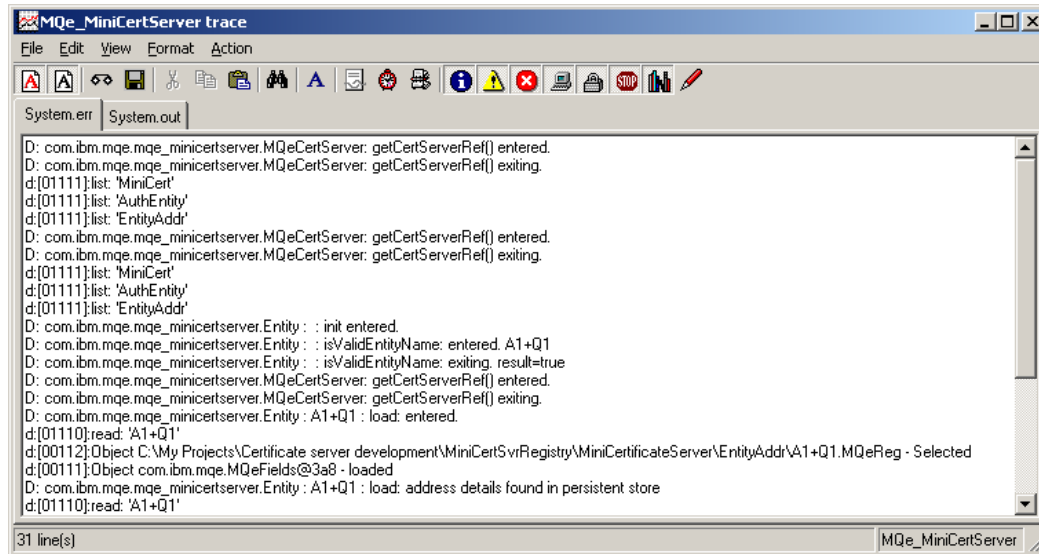


Figure 5-14: The trace window

Menus

The console window shares many of the properties of the main list view pane. Thus columns may be sorted (or re-sorted) by clicking; likewise columns may be re-ordered by dragging their headers. Only one trace window may be present.

File

Log system.err

Log System.err causes the output of *System.err* to be written to a log file in plain text format (.txt). The contents logged are controlled by the settings chosen in the *Action* and *View* menus. The logging is unaffected by stopping and starting the display in the text area, nor does it reflect any changes made to the text area contents.

Save as

Save As saves the entire contents of the uppermost displayed text area (i.e. the output from *System.err* or *System.out*) as either a plain text file (.txt) or as a rich text file (.rtf).

Exit

Exit stops tracing and removes the trace window.

Edit

Cut

Cut cuts the currently selected text in the uppermost displayed text area (i.e. the output from *System.err* or *System.out*) to the system clipboard. The text is stored in both rich text and plain text formats.

Copy

Copy copies the currently selected text in the uppermost displayed text area (i.e. the output from *System.err* or *System.out*) to the system clipboard. The text is stored in both rich text and plain text formats.

Paste

Paste pastes text from the system clipboard to the uppermost displayed text area (i.e. the output from *System.err* or *System.out*). If text at the destination is selected, it is replaced; otherwise the clipboard contents are inserted at the current text cursor position.

Find

Find locates a text string in the uppermost displayed text area (i.e. the output from *System.err* or *System.out*) and, if found, leaves the result selected. If *Find* is invoked with text selected, then that selected string is primed as the search string. All search strings used in an invocation of *Find* are available for re-use in the drop down list box. Repeat identical searches begin where the previous search finished, until all the target text area has been searched. Options are provided for case matched and whole word searches.

Clear

Clear deletes the content of the uppermost displayed text area (i.e. the output from *System.err* or *System.out*).

Delete

Delete deletes the currently selected text in the uppermost displayed text area (i.e. the output from *System.err* or *System.out*) to the system clipboard.

Select all

Select All selects the entire contents of the uppermost displayed text area (i.e. the output from *System.err* or *System.out*).

View

System.err

System.err stops/starts the display of the *System.err* stream in the text area. A timed message is added, indicating when the stream was stopped/started. Note that this message does not appear in any logged output.

System.out

System.out stops/starts the display of the *System.out* stream in the text area. A timed message is added, indicating when the stream was stopped/started.

Thread names

Thread Names determines whether thread names are included in the messages displayed or logged.

Timestamp

Timestamp determines whether timestamps are included in the messages displayed or logged.

Object names

Object Names determines whether object names are included in the messages displayed or logged.

Format

Font

Font formats the currently selected text in the uppermost displayed text area (i.e. the output from *System.err* or *System.out*). The font, style, size, color and effects may be set.

Action

Information

Information causes information trace messages to be captured on *System.err*. *Information* by itself captures application information messages; *Information* with *System* also captures system information messages. If logging is enabled, they will be written to a log file, along with any other captured messages. They are also displayed in the text area under the *System.err* tab, provided that viewing of *System.err* is enabled.

Warning

Warning causes warning trace messages to be captured on *System.err*. *Warning* by itself captures application warning messages; *Warning* with *System* also captures system warning messages. If logging is enabled, they will be written to a log file, along with any other captured messages. They are also displayed in the text area under the *System.err* tab, provided that viewing of *System.err* is enabled.

Error

Error causes error trace messages to be captured on *System.err*. *Error* by itself captures application error messages; *Error* with *System* also captures system error messages; *Error* with *Security* captures system security messages. If logging is enabled, they will be written to a log file, along with any other captured messages. They are also displayed in the text area under the *System.err* tab, provided that viewing of *System.err* is enabled.

System

System causes system trace messages to be captured on *System.err*. It can also be used in conjunction with *Information*, *Warning* and *Error* to capture system information, warning and error messages. If logging is enabled, they will be written to a log file, along with any other captured messages. They are also displayed in the text area under the *System.err* tab, provided that viewing of *System.err* is enabled.

Security

Security causes security trace messages to be captured on *System.err*. *Security* by itself captures application security messages; *Security* with *Error* also captures system security messages. If logging is enabled, they will be written to a log file, along with any other captured messages. They are also displayed in the text area under the *System.err* tab, provided that viewing of *System.err* is enabled..

Exception

Exception causes exception trace messages to be captured on *System.err*. If logging is enabled, they will be written to a log file, along with any other captured messages. They are also displayed in the text area under the *System.err* tab, provided that viewing of *System.err* is enabled.

Debug

Debug causes debug trace messages to be captured on *System.err*. *Debug* by itself captures application debug messages; *Debug* with *System* also captures system debug messages. If logging is enabled, they will be written to a log file, along with any other captured messages. They are also displayed in the text area under the *System.err* tab, provided that viewing of *System.err* is enabled.

Call to debug

Call to Debug causes call to debug trace messages to be captured on *System.err*. If logging is enabled, they will be written to a log file, along with any other captured messages. They are also displayed in the text area under the *System.err* tab, provided that viewing of *System.err* is enabled.

6 CommandConsole: text command user interface

6.1 Outline

Objective

CommandConsole provides a platform neutral, text command-driven interface for the configuration, administration and operational control/monitoring of the certificate issuance server.

General approach

The implementation provides a command shell, which interprets textual user input as commands, and executes those commands to control the certificate issuance server.

Jacl is a port of the Tcl language, implemented in Java. Its strengths lie in being simple, fast, and easy to learn.

The CommandConsole program provided here embeds a Jacl interpreter to which users' commands are sent for execution. Consequently, scripts can be written in the Tcl language, and then executed by the CommandConsole program.

Symbols, notation and formatting of examples

Syntax descriptions for commands available from the CommandConsole program use a representation where:

- `[...]`
The content within the bracket is optional. These can be nested, so if a valid syntax is `a[b[c]]` then `a`, `ab` and `abc` are valid, but `bc`, `c` is not valid.
- `{ ... }`
The content within the bracket contain a list of options, separated by `|` characters. At least one of these options must be used. For example, using a syntax of `a{b|c}` then `ab` and `ac` are valid, but `a`, `abc`, `bc` are invalid.
- `(...)`
The contents within the bracket are grouped, such that their combined value represents an option for the `{ ... }` bracketting above. For example, using a syntax of `a{b|(cd)}` then `ab` or `acd` is valid, but `a`, `ac`, `b`, `cd` is invalid.
- `< ... >`
The contents within the bracket is a symbolic name, the value of which should be substituted with a value provided by the user as the command is typed.
- Code marked in boxes, using a system-proportional font represent things examples which the user types-in at the console, code in files, or syntax descriptions.

this is an example of text marked in this manner.

Resources manipulated

The CommandConsole supports:

- Profiles
- The certificate server
- Entities
- Audit logs
- MQe trace logs

Sources of information on Tcl and Jacl

The web site <http://tcl.activestate.com/doc/> provides a good starting point. Other useful references are:

"Scripting: Higher Level Programming for the 21st Century" - A white paper by John K. Ousterhout : <http://tcl.activestate.com/doc/scripting.html>

List of Tcl faq documents : <http://tcl.activestate.com:8002/resource/doc/faq/>

The Tcl style guide : <http://tcl.activestate.com/doc/styleGuide.pdf>

Brent B Welch : "Practical Programming in Tcl and Tk" ISBN:0-13-022028-0

Starting up the CommandConsole program

From the command line enter one of (or a variation depending upon your JVM):

- Using the Microsoft Java Virtual Machine -
`jview com.ibm.mqe.mqe_minicertserver.CommandConsole`
- Using an IBM Java Virtual Machine -
`java com.ibm.mqe.mqe_minicertserver.CommandConsole`

Once started, you will see a welcome banner from the tool.

```
MQSeries Everyplace - Mini Certificate Issuance Server - Command Console
Version 1.0.0.0
Copyright IBM Corp. 2002. All Rights Reserved
US Government Users Restricted Rights - use, duplication or
disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Try typing 'mqe_cert_help' for information on valid commands.

>
```

The ">" is a prompt for user input. Type your commands whenever you see such a prompt.

Refer to the list of available commands for information on what commands are supported.

Use of parameters when starting the CommandConsole program

Parameters may be specified when the CommandConsole is invoked, with no limit on the number that can be used. Each will be considered to be a line of input to the Jacl interpreter held within the CommandConsole.

Before any user input is obtained from the keyboard, the parameters passed on the command line are processed. For example, if you wish the CommandConsole to execute a Tcl script called *myScript.tcl* at startup, use:

```
java com.ibm.mqe.mqe_minicertserver.CommandConsole "source myScript.tcl"
```

For example, if you wish the CommandConsole to always turn on the audit log and then run the 'myScript.tcl' file when it starts, use:

```
java com.ibm.mqe.mqe_minicertserver.CommandConsole "mqe_cert_auditlog -start" "source myScript.tcl"
```

Alternatively, multiple commands can be grouped into a single invocation parameter in the usual way in which multiple Tcl statements are expressed, using a ';' character between commands:

```
java com.ibm.mqe.mqe_minicertserver.CommandConsole "mqe_cert_auditlog -start ; source myScript.tcl"
```

Once commands passed in this way are completed, the program will wait for further commands on the *stdin* input channel.

Another way of executing a script using the program is by re-directing a script file into the CommandConsole program, as in:

```
cat myScript.tcl | java com.ibm.mqe.mqe_minicertserver.CommandConsole "mqe_cert_auditlog -start"
```

The above command will start the audit log, then run all the Tcl commands from the *myScript.tcl*, and then shut down immediately. This method does not leave the CommandConsole program waiting for user-typed input. In this case, the 'exit' Tcl command is not required to exit the CommandConsole program.

Stopping the CommandConsole program

Stop the CommandConsole program by using the Tcl "exit" command.

Loading and using the mqe_cert package from within a separate Jacl shell

The *com.ibm.mqe.mqe_minicertserver.CommandConsole* is a program in which a Jacl shell is created and used as detailed above.

The same class is also a Jacl extension, such that it can be loaded as a java extension of a Jacl interpreter.

The standard Jacl shell can be invoked, and quite separately, the mqe_cert Jacl extension can be loaded and used in exactly the same manner as if the CommandConsole program were invoked directly.

For example

```
C:\mqe_minicertserver>set CLASSPATH=.\CommandConsole.jar;%CLASSPATH%

C:\mqe_minicertserver>java tcl.lang.Shell
% package names
java Tcl
% package require java
1.2.6
% java::load com.ibm.mqe.mqe_minicertserver.CommandConsole
% package names
java Tcl mqe_cert
% info commands
mqe_cert_server package switch return break subst mqe_cert_trace clock linsert lsort gets
mqe_cert_entity expr variable info tclPkgUnknown vwait for lappend format lrange unset split
regexp exit global list uplevel socket auto_mkindex eof file tclLog regsub case update rename
string set fblocked append cd interp open if auto_execok trace close c
ontinue time read scan pkg_mkIndex fconfigure catch incr foreach eval mqe_cert_profile pwd
unknown join flush array namespace after error llength tell glob lindex proc put
s upvar tclMacPkgSearch lreplace tclPkgSetup auto_load mqe_cert_auditlog seek auto_reset
source concat lsearch while exec mqe_cert_help
%
```

After using the `java::load` command, the `mqe_cert_*` commands are available within this Jacl shell.

6.2 Command reference

mqe_cert_help command

This command displays simple textual help on a list of basic commands for use with your CommandConsole program. The text aims to be a quick reminder of commands you might wish to use, and how to get more information.

Syntax:

mqe_cert_help

Parameters:

No parameters

Returns:

A string containing the help text.

Example of usage

```
>mqe_cert_help
Commands available to control the certificate issuance server are:
mqe_cert_help
  This command. Displays help.
mqe_cert_profile
  Creates or loads a profile.
  Use 'mqe_cert_profile -help' for more information.
mqe_cert_server
  Loads, starts, stops, unloads an issuance server.
  Use 'mqe_cert_server -help' for more information.
mqe_cert_auditlog
  Starts or stops appends to a named audit log file.
  Use 'mqe_cert_auditlog -help' for more information.
mqe_cert_trace
  Starts or stops MQSeries Everyplace trace to a file.
  Use 'mqe_cert_trace -help' for more information.
mqe_cert_entity
  Manipulate entities which request certificates.
  Use 'mqe_cert_entity -help' for more information.
info commands
  Get full list of Jacl commands.
source <file>
  Executes a named Tcl script from a file.
exit
  Exits this program.
or
  Read a book on the Tcl/Tk language
  Search the internet for 'Tcl'
  Refer to the documentation accompanying this program
>
```

mqe_cert_profile command

This command manipulates profiles. On Windows platforms all profiles are held in the Windows registry; on all other platforms each profile is a file on the file system. We recommend that you store all such profile files in a single directory, in a known location on the file system.

Within the CommandConsole, you can use the following standard Tcl commands to navigate around the file system:

```
pwd – displays the name of the current directory.
cd <directory> - moves around the directory structure.
glob * - lists all the files within the current directory.
```

Each profile holds the properties previously described in the ***Error! Reference source not found.*** section on page ***Error! Bookmark not defined.***

The actions applicable to profiles are: create, delete, update and view.

Syntax

To create a profile:

```
mqe_cert_profile –create –passphrase <phrase>
  [-name <filename>] [-maxchannels <number>] [-port <number>]
  [-commsadapter <classname>] [-timeout <seconds>]
  [-storageadapter <classname>] [-registrydir <path>]
  [-auditlogfile <filename>] [-auditlogstart (1|0)]
```

To delete a profile:

```
mqe_cert_profile -delete [-name <filename> ]
```

To update a profile use:

```
mqe_cert_profile –update [-name <filename>] [-maxchannels <number>] [-port <number>]
  [-commsadapter <classname>] [-timeout <seconds>]
  [-auditlogfile <filename>] [-auditlogstart (1|0)]
```

To view the contents of a profile:

```
mqe_cert_profile -view [-name <filename>] [-verbose (1|0)]
```

To get a brief reminder of the syntax of this command:

```
mqe_cert_profile –help
```

Parameters

- **-name**
The name of the profile file to create. The default name is 'default.profile' in the current directory. This can be a fully qualified or relative filename.
- **-passphrase**
A string burnt into the created profile, and into the registry of the certificate server. This value must be supplied to load the certificate issuance server. Keep the value secret and secure to prevent others from loading the certificate server without permission. The passphrase must conform to the rules described in the section *Password and passphrase rules* on page 82.
- **-maxchannels**
Indicates the maximum allowable number of concurrent requests. -1 indicates unlimited. If not specified on the `–create` action, a default value of -1 is assumed.
- **-port**
The IP port used for incoming connection requests. The port number must match that configured in the remote queue manager. If unspecified on the `–create` action, a default of 8085 is assumed.
- **-commsadapter**
The class name of the MQe communications adapter used for the client/server channel from the requesting authenticatable entity. The default value is 'com.ibm.mqe.adapters.MQeTcipHttpAdapter'. This IP adapter must match that configured in the remote queue manager.
- **-storageadapter**
The class name of the MQe storage adapter used to access the registry store. The default value is 'com.ibm.mqe.adapters.MQeDiskFieldsAdapter'.
- **-timeout**
A time in units of one second. After a period of inactivity, with no traffic on a communications channel with a requesting entity, the issuance server closes the channel. Default is 300 seconds.
- **-path**
A directory in which the registry information for this mini certificate server is placed. The certificate issuance server will construct a directory hierarchy below the supplied file path. If this is a relative path, then it will be converted into an absolute path before the profile is stored. If not specified, then the absolute path of the current directory will be used.
- **-verbose**
Indicates whether the `–pindigest` and `–keyringdigest` values are displayed with the `–view` operation.
- **-auditlogfile**
When the server starts, it will collect audit information into this file if the `–auditlogstart` value is '1'. If missing from the `–create` operation, it is defaulted to 'MiniCertServerAuditLog.txt'.
- **-auditlogstart**
A value of '1' indicates that the audit log should be started whenever the certificate server is started, in this case the file used to collect information is specified by the `–auditlogfile` property above. A value of '0' indicates that it should not be started when the server is started. Default is '1'.

Return values

The `-delete`, `-update` and `-create` operations all return an integer value. 1 (the Tcl value for truth) if the operation worked. A *TclException* is thrown with an explanation of the error if these commands do not complete successfully.

The `-help` operation returns a string.

The `-view` operation returns a list of name-value pairs. The name part of these name-value pairs is similar to those defined above, except when the `-verbose` option is used, in which case two extra properties are returned.

- `-passphrase` is not displayed by the `-view` command. This value is never stored persistently and so cannot be returned directly.
- `-passphrasedigest` parameter is a value derived from the original `-passphrase` supplied on a previous `-create` operation.

```
>mqe_cert_profile -view -verbose 1
-name default.profile -maxchannels -1
-commsadapter com.ibm.mqe.adapters.MQeTcipHttpAdapter
-port 8085 -registrydir {C:\certregistry}
-storageadapter com.ibm.mqe.adapters.MQeDiskFieldsAdapter -timeout 300
-passphrasedigest 5DDF3502B217BC6A758E1A1E2421B719B8A8689B
-auditlogfile MiniCertServerAuditLog.txt -auditlogstart 1
>
```

Examples

Create a default profile, such that the `-pin` and `-keyring` passwords are '12345678', then view it's values.

```
>mqe_cert_profile -create -passphrase 1234567812345678
1
>
```

Create a profile called 'C:\my profiles\profile1.profile', such that the port used is 8086, and channels timeout after 20 seconds.

```
>mqe_cert_profile -create -name "C:/my profiles/profile1.profile" -port 8086 -maxchannels 20
-passphrase 1234567812345678
1
>
```

Note: In Tcl, `\` is an escape character. Use `'` instead or use `\\` where `\` would normally exist in the path name.

Note: Tcl brackets " ... " group characters together, and can be helpfully used to allow spaces in filenames.

Update that same profile to make the server use port 8087 instead.

```
>mqe_cert_profile -update -name {C:\\my profiles\\profile1.profile} -port 8087
1
>
```

Note: In Tcl, `{ ... }` is another grouping mechanism in which spaces between words are not split up. Again, this can be helpfully used to allow spaces in filenames.

Delete the profile created above

```
>mqe_cert_profile -delete -name {C:/my profiles/profile1.profile}
1
>
```

mqe_cert_server command

This command controls the certificate issuance server.

State model used by an issuance server

The server has several states:

- *unloaded*

The server is not loaded. No manipulation of entities or certificates, or issuance of certificates is possible.

The `-load` operation will load a server based on values defined in a profile at the time of loading. Depending on the state of the `-auditlogstart` flag, the server may or may not start collecting details of important events in the audit log file. See the documentation of the `mqe_cert_profile` command for more details.

This moves the server to the *loaded* state.

The `-passphrase` value is required to load the server.

- *loaded*

When in this state, the administration of requesting entities and their certificates is allowed, though entities requesting certificates will be refused.

Only one server can be loaded at a time inside a single CommandConsole shell.

Using the `-unload` operation moves the server back to the *unloaded* state.

Using the `-start` operation moves the server to the *running* state.

- *running*

The certificate issuance server allows administration of requesting entities and their certificates is allowed, and the issuance server may also issue certificates to requesting entities.

The server can be stopped, returning it to the *loaded* state where it will refuse requests for certificates, by using the `-stop` operation.

At any point, characteristics of the current server can be viewed using the `-view` operation.

Syntax

To view the current state of the server in the CommandConsole:

```
mqe_cert_server -view
```

To move the server from *unloaded* to *loaded* state:

```
mqe_cert_server -load -passphrase <phrase> [-name <filename>]
```

To move the server from *loaded* to *running* state:

```
mqe_cert_server -start
```

To move the server from *running* to *loaded* state:

```
mqe_cert_server -stop
```

To move the server from *loaded* to *unloaded* state:

```
mqe_cert_server -unload
```

To get a brief reminder of the syntax for the `mqe_cert_server` command:

```
mqe_cert_server -help
```

Parameters

- `-passphrase`
The secret string value¹⁴ that was originally specified when the profile was created previously with the `mqe_cert_profile` command. The value specified must match that supplied when the profile was created, or the `-load` command will fail.
- `-name`
The filename referring to a profile file previously created using the `mqe_cert_profile` command (see above for details of that command). If not specified, a default value of 'default.profile' will be used, from the current directory.

Return values

The `-view` operation returns a list of name-value pairs; for example:

```
>mqe_cert_server -view
-state running -port 8085 -maxchannels -1 -timeout 300
-commsadapter com.ibm.mqe.adapters.MQeTcipHttpAdapter
-storageadapter com.ibm.mqe.adapters.MQeDiskFieldsAdapter
-registrydir {C:\myRegistryLocation}
>
```

The `-state` property reflects the state of the server, *unloaded*, *loaded* or *running*.

The `-help` operation returns a string with help text included. The `-load` `-start` `-stop` and `-unload` operations all return a Tcl "truth" value of 1, or throw a *TclException*, with a text message indicating why the operation failed.

¹⁴ We do not recommend that you hard-code password values in your scripts, but prompt for them using the Tcl 'gets' command.

Examples

Load a server using the default profile using a command typed at the console and start the server issuing certificates; then display the state of the server:

```
>mqe_cert_server -load -passphrase 1234567812345678
1
>mqe_cert_server -start
1
>mqe_cert_server -view
-state running -port 8085 -maxchannels -1 -timeout 300
-commsadapter com.ibm.mqe.adapters.MQeTcipHttpAdapter
-storageadapter com.ibm.mqe.adapters.MQeDiskFieldsAdapter
-registrydir {C:\myRegistryLocation}
>
```

Stop and unload the server, then display the state of the server:

```
>mqe_cert_server -stop
1
>mqe_cert_server -unload
1
>mqe_cert_server -view
-state unloaded
>
```

mqe_cert_auditlog command

The audit log is a file containing a log of all the major events that occurred between the time when the logging was started and when it stopped. Use this command to start and stop the collection of audit log information.

The audit log cannot be started until the certificate server is in loaded state. When the certificate server is initially loaded, the audit log may or may not be started based on the `-auditlogstart` and `-auditlogfile` parameters in the profile (see documentation of the `mqe_cert_profile` command for more details of these parameters).

Syntax

To start the audit log capturing information:

```
mqe_cert_auditlog -start [-file <filename>]
```

To stop the audit log capturing information:

```
mqe_cert_auditlog -stop
```

To display help text on the command:

```
mqe_cert_auditlog -help
```

To display the state of the audit log, i.e. running or stopped; if running, the identity of the log file:

```
mqe_cert_auditlog -view
```

Parameters

- `-file`
The name of the file in which audit log information will be placed. If the file does not exist, it will be created; if the file already exists, log data will be appended to it. A filename of 'MiniCertServerAuditLog.txt' is used by default, with the file being created in the current directory.

Return values

The `-start` and `-stop` operations return the Tcl truth value "1" if successful; otherwise an exception description in the event of a failure.

Examples of usage

Tell the server to capture major events into the default audit log filename, in the current directory:

```
>mqe_cert_auditlog -start
1
># load the server
>mqe_cert_server -load -pin 12345678 -keyring 12345678
1
># allow entity 'myQueueManager' to request a certificate.
>mqe_cert_entity -create -name myQueueManager -reqpin 12345678
1
```

Will produce an extra line in the *MiniCertServerAuditLog.txt* file:

```
15-Mar-02 09:49:34 :Set request pin. Requesting entity name:'myQueueManager'
Pin:'12345678' Time:'8640' Tries:'3'
```

Example showing a use of the `-view` operation:

```
>mqe_cert_auditlog -start
1
>mqe_cert_auditlog -view
-state 1 -file MiniCertServerAuditLog.txt
>mqe_cert_auditlog -stop
1
>mqe_cert_auditlog -view
-state 0
>
```

mqe_cert_trace command

This command is used to gather information for IBM service personnel, in the event that there is a problem with the certificate issuance server code.

If a member of IBM staff instruct you to gather a detailed trace of activity within the server, use this command to have the information collected and deposited into a file. The file can then be sent to the IBM.

Syntax

To start capturing trace information to a file:

```
mqe_cert_trace -start [-file <filename>]
```

To stop the of trace information:

```
mqe_cert_trace -stop
```

To view help text on the command

```
mqe_cert_trace -help
```

To view whether trace is running or stopped and, if running, which file is being used.

```
mqe_cert_trace -view
```

Parameters

- **-file**
The name of the file in which trace information will be placed. If the file does not exist, it will be created. If the file already exists, it will be overwritten. A default filename of MiniCertServerTrace.txt is created in the current directory.

Return values

The `-start` and `-stop` operations return the Tcl value for truth, a numeric 1. The `-help` command returns a string describing the syntax of the command.

Examples of usage

Tell the server to start capturing traced events into the default trace filename, load the server, then stop the capture of trace:

```
>mqe_cert_trace -start
1
># load the server
>mqe_cert_server -load -pin 12345678 -keyring 12345678
1
>mqe_cert_trace -stop
1
```

The MiniCertServerTrace.txt file can then be sent to IBM.

Example of a use of the –view operation:

```
>mqe_cert_trace -start  
1  
>mqe_cert_trace -view  
-state 1 -file MiniCertServerTrace.txt  
>mqe_cert_trace -stop  
1  
>mqe_cert_trace -view  
-state 0  
>
```

mqe_cert_entity command

This command manipulates entities known to the certificate issuance server. It allows you to create, update, view or delete entities in the persistent registry controlled by the issuance server.

Syntax

To create an entity:

```
mqe_cert_entity -create -name <entityname>
  [ -address0 <detail> ][ -address1 <detail> ][ -address2 <detail> ][ -address3 <detail> ]
  [ ( (-pinstate unavailable)
    | ( [-pinstate available] [ -reqpin <password> [-tries <number>] [-months <number>]] )
  ) ]
```

To get a list of names of all entities which currently exist:

```
mqe_cert_entity -list
```

To view all properties of a named entity:

```
mqe_cert_entity -view -name <entityname> [-verbose (1|0)]
```

To remove an entity¹⁵:

```
mqe_cert_entity -delete -name <entityname>
```

To update an existing entity:

```
mqe_cert_entity -update -name <entityname>
  [ -address0 <detail> ][ -address1 <detail> ][ -address2 <detail> ][ -address3 <detail> ]
  [ ( [-pinstate unavailable]
    | ( [-pinstate available] [ -reqpin <password> [-tries <number>] [-months <number>]] )
  ) ]
```

To obtain simple help on the command:

```
mqe_cert_entity -help
```

¹⁵ The certificate issuance server's entity cannot be deleted.

Parameters

- **-address0 ... address3**
A general purpose string in which information pertaining to the entity can be stored. The meaning of each value is up to the user of the CommandConsole program.
- **-name**
The name of the entity. For a queue manager entity, this is the name of an MQE queue manager; for a queue entity, this is of the form: <queue-manager-name>+<queue-name>.
- **-pinstate**
A state which reflects the authority the entity has to request a certificate or not
 - **available**
Indicates that if the entity requests a certificate, one will be issued. Using this option requires that you also specify the `-reqpin` option.
 - **unavailable**
Indicates that if the entity requests a certificate, one will NOT be issued. Use of this option is incompatible with a use of the `-reqpin` option.

The default `-pinstate` is assumed to be *available* if the `-reqpin` option is present, otherwise a `-pinstate` of *unavailable* is assumed.
- **-tries**
The number of tries a requesting entity is allowed to match their request pin with that already associated with the entity. Each request for a certificate where these passwords don't match will decrement the number of permitted tries remaining. When the number reaches zero, the pinstate changes from *available* to *unavailable*.
- **-months**
The number of months from the time a certificate is issued, for which the certificate will remain valid.
- **-verbose**
Indicates how much detail should be returned as a result of the `-view` operation. If set to true "1" then all details about the entity, and any certificate owned by that entity will be displayed. If set to false, "0", then only a subset of that information is returned. A default of "0" is assumed if the `-verbose` parameter is not specified. It is not normally expect that anyone, except those debugging their security application, would look at the extra information provided by using this flag.

Return values

The `-create` `-update` and `-delete` operations all return the Tcl truth value '1' if the operation is successful, otherwise a Tcl exception which explains the reason for the failure will be returned.

The `-list` operation returns an unordered list of the entity names known to the certificate server.

The `-view` operation returns a list of name-value pairs. These reflect the properties of the entity being viewed. The properties are the same as those used for the `-create` operation, with some additions.

- `-notafter`
The date after which the certificate is unusable.
- `-notbefore`
The date before which the certificate is unusable.
- `-certversion`
The version of the certificate standard used to create the certificate.
- `-issuer`
The name of the issuing entity, plus any attributes.
- `-signature`
The digital signature.
- `-subject`
The name of the entity who requested the certificate, plus any attributes.
- `-algorithm`
An indicator of which algorithm was used to generate the signature.
- `-publickeytype`
- `-paramspecifier`
The algorithm parameter specifier.
- `-rsakeymodulus`
The modulus element of the public key.
- `-rsakeyexponent`
The exponent element of the public key.

Examples of usage

List available entities, create another entity 'myQueueManager', then list the available entities again:

```
>mqe_cert_entity -list

>mqe_cert_entity -create -name myQueueManager -address0 {10 Downing Street}
-address1 {Westminster} -address2 {London} -address3 {United Kingdom}
-reqpin 12345678
1
>mqe_cert_entity -list
myQueueManager
>mqe_cert_entity -view -name myQueueManager
-name myQueueManager -pinstate available -tries 3 -months 12
-address0 {10 Downing Street} -address1 Westminster -address2 London
-address3 {United Kingdom}
>
```

Once the queue manager entity has been created, and the “myQueueManager” has successfully requested a certificate, the view of the entity changes:

```
># View the entity
>mqe_cert_entity -view -name myQueueManager
-name myQueueManager -pinstate used -address0 {10 Downing Street}
-address1 Westminster -address2 London -address3 {United Kingdom}
-notbefore {16 March 2002 20:03} -notafter {16 March 2003 20:03}
># View the entity is verbose mode.
>mqe_cert_entity -view -name myQueueManager -verbose 1
-name myQueueManager -pinstate used -address0 {10 Downing Street}
-address1 Westminster -address2 London -address3 {United Kingdom}
-notbefore {16 March 2002 20:03} -notafter {16 March 2003 20:03}
-certversion 1 -issuer {MQeMCS@COBBETTM; ; } -subject {myQueueManager; ; }
-algorithm 2 -publickeytype 2 -paramspecifier 0
-signature
3C38E090795AFB1A41405FCF4AFFDA07DDBD2134D8F2878A1CAECDA16B564DFDF7BC7BA6D4E
3FF031E3ED0AACF23550F9E80087F293B652447AC8C9921B7FD710ED16CEED8A0D54A9F433CA
ACC3B71C4785DADE107E59D2FCD4375088A55C579353614DFACD2FD2B1B6
6758C333CE05403622F7AC449CDBFA410CCCE3334FA4C
-rsakeyexponent 010001
-rsakeymodulus
00C6E45DEBD53347509DBC5869AE2CEADA540AF131EFF5DE6BE90AEAE54CAEE5BF755ED89D
C5BA40C40BD270AE98EEBDA73685C83EF675AF05A4E3D46BD038FEE60CC59983DDDF22709
39DD0A6745229873F206EC1677DF33F79DE08E11EF7A385FCCFFA143B0FFE2EB03CF75E8210FD
7B5F25128593C2B9E17E2
2C61C5B4BD
```

Note the –notbefore and –notafter values indicate when the validity period for the certificate.

6.3 Scripting using Tcl and the *mqe_** commands

The availability of Tcl commands within the CommandConsole provide a useful set of scripting facilities.

The package name of *mqe_cert* has been registered with the Tcl interpreter, such that the version holds 1.0.0.0 of the Tcl code.

The standard mechanism for verifying that a package is present in the interpreter is recommended. For example:

```
>package names
java Tcl mqe_cert
>package require mqe_cert
1.0.0.0
>
```

To display list items in a different way, use the `foreach` Tcl command. For example, to view the list of entity names, such that each list item appears on a separate output line, use:

```
>foreach {entityName} [mqe_cert_entity -list] {
>>puts "$entityName"
>>}
```

For example, to display name-value pairs, one per line, use:

```
>foreach {name value} [mqe_cert_entity -view -name myTestQMgr] {
>>puts "$name $value"
>>}
-name myTestQMgr
-pinststate available
-tries 3
-months 12
-address0 MCS queue manager test
-address1 IBM
-address2 Hursley Park
-address3 Winchester
>
```

To filter the entity names such that only “my...” entities are displayed:

```
>foreach {entityName} [mqe_cert_entity -list] {
>> if { [string match {my*} "$entityName"] } {
>> puts "$entityName"
>> }
>>}
myTestQMgr+myTestQ
myTestQMgr
>
```

Errors reported by the *mqe_cert** commands are thrown as tcl exceptions. (as the Tcl Error command will do). These can be converted into a 1 (failure) or 0 (success) using the Tcl catch command.

For example:

```

if { [ catch { mqe_cert_server -load -name $profile_name -passphrase $passphrase }
      error_info ] } {
    puts "The server failed to load: $error_info";
    ... recover from the error...
} else {
    ... do the things which can only be done if the server load works...
}

```

A sample script `mqe_cert.tcl` is also shipped in this supportpac. It provides some potentially useful commands you can also use in your scripts.

To use this tcl file, from inside the interpreter, use the Tcl “source” command. as in:

```

>source mqe_cert.tcl
You have the following mqe_cert commands available:
mqe_cert_auditlog
mqe_cert_entity
mqe_cert_help
mqe_cert_is_server_running
mqe_cert_list_cert_commands
mqe_cert_list_entities
mqe_cert_list_profiles
mqe_cert_profile
mqe_cert_server
mqe_cert_start_serving
mqe_cert_trace
>

```







As you can see, the script helpfully lists the set of `mqe_cert*` commands available after the script has run.

7 Appendix A: Reference section

7.1 Icons and buttons







MQe_MiniCertServer uses the following icons and buttons (where two icons are shown the first represents the unselected icon, the second the selected icon).

Icons







		MQe root (MQe_MiniCertServer)
		Queue manager entity
		Queue entity

Buttons





















Toolbar – main window

	Close current profile (≡ <i>File</i> → <i>Close</i>)
	Create a new profile (≡ <i>File</i> → <i>New Queue Mgr.</i>)
	Open an existing profile (≡ <i>File</i> → <i>Open</i>)
	Refresh selected object (≡ <i>View</i> → <i>Refresh</i>)
	Start serving certificates (≡ <i>Action</i> → <i>Start</i>)
	Stop serving certificates (≡ <i>Action</i> → <i>Stop</i>)

Toolbar – report window

	Copy selected tab contents to a file (≡ <i>File</i> → <i>Save As...</i>)
	Cut selected text to the clipboard (≡ <i>Edit</i> → <i>Cut</i>)
	Copy selected text to the clipboard (≡ <i>Edit</i> → <i>Copy</i>)
	Paste text from the clipboard (≡ <i>Edit</i> → <i>Paste</i>)
	Find text string (≡ <i>Edit</i> → <i>Find</i>)
	Change font formatting of selected text (≡ <i>Format</i> → <i>Font</i>)

Toolbar – trace window

	Copy System.err to a log file (≡ <i>File</i> → <i>Log System.err</i>)
	Copy selected tab contents to a file (≡ <i>File</i> → <i>Save As...</i>)
	Cut selected text to the clipboard (≡ <i>Edit</i> → <i>Cut</i>)
	Copy selected text to the clipboard (≡ <i>Edit</i> → <i>Copy</i>)
	Paste text from the clipboard (≡ <i>Edit</i> → <i>Paste</i>)
	Find text string (≡ <i>Edit</i> → <i>Find</i>)
	Change font formatting of selected text (≡ <i>Format</i> → <i>Font</i>)
	Include time stamps (≡ <i>View</i> → <i>Timestamps</i>)
	Include object names (≡ <i>View</i> → <i>Object Names</i>)
	Include thread names (≡ <i>View</i> → <i>Thread Names</i>)
	Show calls to debug messages (≡ <i>Action</i> → <i>Calls to Debug</i>)
	Show debug messages (≡ <i>Action</i> → <i>Debug</i>)
	Show error messages (≡ <i>Action</i> → <i>Error</i>)
	Show exception messages (≡ <i>Action</i> → <i>Exception</i>)
	Show information messages (≡ <i>Action</i> → <i>Information</i>)
	Show security messages (≡ <i>Action</i> → <i>Security</i>)
	Show system messages (≡ <i>Action</i> → <i>System</i>)
	Show the contents of System.err (≡ <i>View</i> → <i>System.err</i>)
	Show the contents of System.out (≡ <i>View</i> → <i>System.out</i>)
	Show warning messages (≡ <i>Action</i> → <i>Warning</i>)

7.2 Accessibility features

The following general keystroke aids to navigation are supported:

Keystroke(s)	Place	Function
↑	Main window: list pane	Go up one list view item
↑	All windows: menu	Go up one menu item
↑	Main window: tree pane	Go up one node in the tree
↓	Main window: list pane	Go down one list view item If nothing is selected, select the first item
↓	All windows: menu	Go down one menu item
↓	Main window: tree pane	Go down one node in the tree
←	Main window: list pane	Scroll left
←	All windows: menu	Go left one menu item
←	Main window: tree pane	Shrink current node in the tree
→	Main window: list pane	Scroll right
→	All windows: menu	Go right one menu item
→	Main window: tree pane	Expand the current node in the tree
F6	Main window	Cycle through panes
F10	All windows	Move focus to the menu bar <i>(then press underlined key to activate menu item)</i>
Alt	All windows	Move focus to the menu bar <i>(then press underlined key to activate menu item)</i>
Alt+space	All windows	Control box
Alt+tab	All	Cycle through windows
AltGr+tab	Trace Property pages	Cycle through tabs
Ctrl +tab	Trace Property pages	Cycle through tabs
Tab	Property pages	Cycle forwards through input fields & buttons
Enter	Property pages: buttons	Click selected button
Enter	All windows: menus	Click selected menu item
Shift+Tab	Property pages	Cycle backwards through input fields & buttons

Figure 7-1: General keystroke aids to navigation

The following keystroke aids to navigation are supported only in the Trace window *System.err* and *System.out* text areas:

Keystroke(s)	Function
↑	Move insertion point up a line
↓	Move insertion point down a line
←	Move insertion point one character to the left
→	Move insertion point one character to the right
Ctrl+↑	Move insertion point one line up
Ctrl+↓	Move insertion point one line down
Ctrl+←	Move insertion point one character to the left
Ctrl+→	Move insertion point one character to the right
Ctrl+Shift+↑	Extend selection to the beginning of the paragraph
Ctrl+Shift+↓	Extend selection to the end of the paragraph
Ctrl+Shift+←	Extend selection to the beginning of the word
Ctrl+Shift+→	Extend selection to the end of the word
Ctrl+A	Extend selection to include all text
Ctrl+End	Move insertion point to the end
Ctrl+Home	Move insertion point to the beginning
End	Move insertion point to the end of the line
Home	Move insertion point to the beginning of the line
Shift+↑	Extend selection one line up
Shift+↓	Extend selection one line down
Shift+←	Extend selection one character to the left
Shift+→	Extend selection one character to the right
Ctrl+Shift+Home	Extend selection to the beginning
Ctrl+Shift+End	Extend selection to the end
Shift+End	Extend selection to the end of the line
Shift+Home	Extend selection to the beginning of the line
Shift+PgDn	Extend selection to end
Shift+PgUp	Extend selection to beginning
Tab	If no insertion point, display insertion point at end If insertion point present, insert a tab

Figure 7-2: Restricted keystroke aids to navigation

Shortcuts

The following keystroke shortcuts are supported:

Keystroke(s)	Window	Name	Menu equivalent	Function
Ctrl+A	Main Report Trace	Select All	Edit→Select All	Select all
Ctrl+C	Main Report Trace	Copy	Edit→Copy	Copy to clipboard
Ctrl+Delete	Report Trace	Clear	Edit→Clear	Clear
Ctrl+F	Report Trace	Find	Edit→Find	Find string/word
Ctrl+O	Main	Open	File→Open	Open a profile
Ctrl+P	Main	Stop	Action→Stop	Stop serving certificates
Ctrl+S	Main	Start	Action→Start	Start serving certificates
Ctrl+X	Report Trace	Cut	Edit→Cut	Cut selected text to clipboard
Ctrl+V	Report Trace	Paste	Edit→Paste	Paste text from clipboard
Delete	Main Report Trace	Delete	Edit→Delete	Delete selected text
F4	Main	Properties	File→Properties	Display selected object properties
F5	Main	Refresh	View→Refresh	Refresh selected object

Figure 7-3: Keystroke shortcuts

7.3 Password and passphrase rules

MCS passwords and passphrases must not be trivial or obvious. Passwords must be at least six characters long and contain at least four different characters; passphrases must be at least ten characters long and contain at least six different characters.

8 Appendix B: Notices

This information was developed for products and services offered in the U.S.A. IBM® may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
 IBM Corporation
 North Castle Drive
 Armonk, NY 10504-1785
 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM United Kingdom Laboratories,
 Mail Point 151,
 Hursley Park,
 Winchester,
 Hampshire
 England
 SO21 2JN

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee. The licensed program described in this information and all licensed material

available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of International Business machines Corporation in the United States, or other countries, or both.

AIX

IBM

MQSeries

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and/or other countries.

UNIX is a registered trademark of X/Open in the United States and other countries.

Windows and Windows NT are registered trademarks of Microsoft Corporation in the United States and other countries.

WAP Forum is a registered trademark of the Wireless Application Protocol Forum Ltd.

Other company, product, and service names may be trademarks or service marks of others.

9 Appendix C: International Program License Agreement

Part 1 - General terms

PLEASE READ THIS AGREEMENT CAREFULLY BEFORE USING THE PROGRAM. IBM WILL LICENSE THE PROGRAM TO YOU ONLY IF YOU FIRST ACCEPT THE TERMS OF THIS AGREEMENT. BY USING THE PROGRAM YOU AGREE TO THESE TERMS. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE UNUSED PROGRAM TO THE PARTY (EITHER IBM OR ITS RESELLER) FROM WHOM YOU ACQUIRED IT TO RECEIVE A REFUND OF THE AMOUNT YOU PAID.

The Program is owned by International Business Machines Corporation or one of its subsidiaries (IBM) or an IBM supplier, and is copyrighted and licensed, not sold.

The term "Program" means the original program and all whole or partial copies of it. A Program consists of machine-readable instructions, its components, data, audio-visual content (such as images, text, recordings, or pictures), and related licensed materials.

This Agreement includes Part 1 - General Terms, Part 2 - Country-unique Terms, and "License Information" and is the complete agreement regarding the use of this Program, and replaces any prior oral or written communications between you and IBM. The terms of Part 2 and License Information may replace or modify those of Part 1.

1. License

Use of the Program

IBM grants you a nonexclusive license to use the Program.

You may 1) use the Program to the extent of authorizations you have acquired and 2) make and install copies to support the level of use authorized, providing you reproduce the copyright notice and any other legends of ownership on each copy, or partial copy, of the Program.

If you acquire this Program as a program upgrade, your authorization to use the Program from which you upgraded is terminated.

You will ensure that anyone who uses the Program does so only in compliance with the terms of this Agreement.

You may not 1) use, copy, modify, or distribute the Program except as provided in this Agreement; 2) reverse assemble, reverse compile, or otherwise translate the Program except as specifically permitted by law without the possibility of contractual waiver; or 3) sublicense, rent, or lease the Program.

Transfer of Rights and Obligations

You may transfer all your license rights and obligations under a Proof of Entitlement for the Program to another party by transferring the Proof of Entitlement and a copy of this Agreement and all documentation. The transfer of your license rights and obligations terminates your authorization to use the Program under the Proof of Entitlement.

2. Proof of Entitlement

The Proof of Entitlement for this Program is evidence of your authorization to use this Program and of your eligibility for warranty services, future upgrade program prices (if announced), and potential special or promotional opportunities.

3. Charges and Taxes

IBM defines use for the Program for charging purposes and specifies it in the Proof of Entitlement. Charges are based on extent of use authorized. If you wish to increase the extent of use, notify IBM or its reseller and pay any applicable charges. IBM does not give refunds or credits for charges already due or paid.

If any authority imposes a duty, tax, levy or fee, excluding those based on IBM's net income, upon the Program supplied by IBM under this Agreement, then you agree to pay that amount as IBM specifies or supply exemption documentation.

4. Limited Warranty

IBM warrants that when the Program is used in the specified operating environment it will conform to its specifications. IBM does not warrant uninterrupted or error-free operation of the Program or that we will correct all Program defects. You are responsible for the results obtained from the use of the Program. The warranty period for the Program expires when its Program services are no longer available. The License Information specifies the duration of Program services.

During the warranty period warranty service is provided without charge for the unmodified portion of the Program through defect-related Program services. Program services are available for at least one year following the Program's general availability. Therefore, the duration of warranty service depends on when you obtain your license. If the Program does not function as warranted during the first year after you obtain your license and IBM is unable to resolve the problem by providing a correction, restriction, or bypass, you may return the Program to the party (either IBM or its reseller) from whom you acquired it and receive a refund in the amount you paid for it. To be eligible, you must have acquired the Program while Program services (regardless of the remaining duration) were available for it.

THESE WARRANTIES ARE YOUR EXCLUSIVE WARRANTIES AND REPLACE ALL OTHER WARRANTIES OR CONDITIONS, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

These warranties give you specific legal rights, and you may also have other rights which vary from jurisdiction to jurisdiction. Some jurisdictions do not allow the exclusion or limitation of implied warranties, so the above exclusion or limitation may not apply to you. In that event such warranties are limited in duration to the warranty period. No warranties apply after that period.

5. Limitation of Liability

Circumstances may arise where, because of a default on IBM's part or other liability, you are entitled to recover damages from IBM. In each such instance, regardless of the basis on which you may be entitled to claim damages from IBM, (including fundamental breach, negligence, misrepresentation, or other contract or tort claim), IBM is liable for no more than 1) damages for bodily injury (including death) and damage to real property and tangible personal property and 2) the amount of any other actual direct damages up to the greater of U.S. \$100,000 (or equivalent in your local currency) or the charges for the Program that is the subject of the claim.

IBM WILL NOT BE LIABLE FOR ANY SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES (INCLUDING LOST PROFITS OR SAVINGS), EVEN IF IBM, OR ITS RESELLER, HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

IBM will not be liable for 1) loss of, or damage to, your records or data or 2) any damages claimed by you based on any third party claim.

This limitation of liability also applies to any developer of a Program supplied to IBM. It is the maximum for which IBM and its suppliers are collectively responsible.

6. General

Nothing in this Agreement affects any statutory rights of consumers that cannot be waived or limited by contract.

IBM may terminate your license if you fail to comply with the terms of this Agreement. If IBM does so, your authorization to use the Program is also terminated.

You agree to comply with applicable export laws and regulations.

Neither you nor IBM will bring a legal action under this Agreement more than two years after the cause of action arose unless otherwise provided by local law without the possibility of contractual waiver or limitation.

Neither you nor IBM is responsible for failure to fulfill any obligations due to causes beyond its control.

The laws of the country in which you acquire the Program govern this Agreement, except 1) in Australia, the laws of the State or Territory in which the transaction is performed govern this Agreement; 2) in Albania, Armenia, Belarus, Bosnia/Herzegovina, Bulgaria, Croatia, Czech Republic, Georgia, Hungary, Kazakhstan, Kirghizia, Former Yugoslav Republic of Macedonia (FYROM), Moldova, Poland, Romania, Russia, Slovak Republic, Slovenia, Ukraine, and Federal Republic of Yugoslavia, the laws of Austria govern this Agreement; 3) in the United Kingdom, all disputes relating to this Agreement will be governed by English Law and will be submitted to the exclusive jurisdiction of the English courts; 4) in Canada, the laws in the Province of Ontario govern this Agreement; and 5) in the United States and Puerto Rico, and People's Republic of China, the laws of the State of New York govern this Agreement.

Part 2 - Country-unique terms

AUSTRALIA:

Limited Warranty (Section 4):

The following paragraph is added to this Section:

The warranties specified in this Section are in addition to any rights you may have under the Trade Practices Act 1974 or other legislation and are only limited to the extent permitted by the applicable legislation.

Limitation of Liability (Section 5):

The following paragraph is added to this Section:

Where IBM is in breach of a condition or warranty implied by the Trade Practices Act 1974, IBM's liability is limited to the repair or replacement of the goods, or the supply of equivalent goods. Where that condition or warranty relates to right to sell, quiet possession or clear title, or the goods are of a kind ordinarily acquired for personal, domestic or household use or consumption, then none of the limitations in this paragraph apply.

EGYPT:

Limitation of Liability (Section 5):

The following replaces item 2 in the first paragraph of this Section:

2) as to any other actual direct damages, IBM's liability will be limited to the total amount you paid for the Program that is the subject of the claim.

FRANCE :

Limitation of Liability (Section 5):

The following replaces the second sentence in the first paragraph of this Section:

In such instances, regardless of the basis on which you are entitled to claim damages from IBM, IBM is liable for no more than 1) damages for bodily injury (including death) and damage to real property and tangible personal property; and 2) the amount of any other actual direct damages up to the greater of a) U.S. \$100,000 (or equivalent in local currency) or b) the charges for the Program which is the subject of the claim.

GERMANY:

Limited Warranty (Section 4):

The following paragraphs are added to this Section:

The minimum warranty period for Programs is six months.

In case a Program is delivered without Specifications, we will only warrant that the Program information correctly describes the Program and that the Program can be used according to the Program information. You have to check the usability according to the Program information within the "money-back guaranty" period.

The following replaces the first sentence of the first paragraph of this Section:

The warranty for an IBM Program covers the functionality of the Program for its normal use and the Program's conformity to its Specifications.

Limitation of Liability (Section 5):

The following paragraph is added to the Section:

The limitations and exclusions specified in the Agreement will not apply to damages caused by IBM with fraud or gross negligence, and for express warranty.

In item 2, replace "U.S. \$100,000" with "DEM 1.000.000".

The following sentence is added to the end of item 2 of the first paragraph:

IBM's liability under this item is limited to the violation of essential contractual terms in cases of ordinary negligence.

INDIA:

Limitation of Liability (Section 5):

The following replaces items 1 and 2 in the first paragraph:

1) liability for bodily injury (including death) or damage to real property and tangible personal property will be limited to that caused by IBM's negligence; and 2) as to any other actual damage arising in any situation involving nonperformance by IBM pursuant to, or in any way related to the subject of this Agreement, IBM's liability will be limited to the charge paid by you for the individual Program that is the subject of the claim.

General (Section 6):

The following replaces the fourth paragraph of this Section:

If no suit or other legal action is brought, within two years after the cause of action arose, in respect of any claim that either party may have against the other, the rights of the concerned party in respect of such claim will be forfeited and the other party will stand released from its obligations in respect of such claim.

IRELAND:

Limited Warranty (Section 4):

The following paragraph is added to this Section:

Except as expressly provided in these terms and conditions, all statutory conditions, including all warranties implied, but without prejudice to the generality of the foregoing, all warranties implied by the Sale of Goods Act 1893 or the Sale of Goods and Supply of Services Act 1980 are hereby excluded.

Limitation of Liability (Section 5):

The following replaces items 1 and 2 in the first paragraph of this Section:

1) death or personal injury or physical damage to your real property solely caused by IBM's negligence; and 2) the amount of any other actual direct damages, up to the greater of Irish Pounds 75,000 in respect of Programs or 125 percent of the charges for the Program that is the subject of the claim or which otherwise gives rise to the claim.

The following paragraph is added at the end of this Section:

IBM's entire liability and your sole remedy, whether in contract or in tort, in respect of any default will be limited to damages.

ITALY:

Limitation of Liability (Section 5):

The following replaces the second sentence in the first paragraph:

In each such instance unless otherwise provided by mandatory law, IBM is liable for no more than damages for bodily injury (including death) and damage to real property and tangible personal property and 2) as to any other actual damage arising in all situations involving non-performance by IBM pursuant to, or in any way related to the subject matter of this Agreement, IBM's liability, will be limited to the total amount you paid for the Program that is the subject of the claim.

NEW ZEALAND:

Limited Warranty (Section 4):

The following paragraph is added to this Section:

The warranties specified in this Section are in addition to any rights you may have under the Consumer Guarantees Act 1993 or other legislation which cannot be excluded or limited. The Consumer Guarantees Act 1993 will not apply in respect of any goods or services which IBM provides, if you require the goods or services for the purposes of a business as defined in that Act.

Limitation of Liability (Section 5):

The following paragraph is added to this Section:

Where Programs are not acquired for the purposes of a business as defined in the Consumer Guarantees Act 1993, the limitations in this Section are subject to the limitations in that Act.

PEOPLE'S REPUBLIC OF CHINA:

Charges (Section 3):

The following paragraph is added to the Section:

All banking charges incurred in the People's Republic of China will be borne by you and those incurred outside the People's Republic of China will be borne by IBM.

UNITED KINGDOM:

Limitation of Liability (Section 5):

The following replaces items 1 and 2 in the first paragraph of this Section:

1) death or personal injury or physical damage to your real property solely caused by IBM's negligence; 2) the amount of any other actual direct damages, up to the greater of Pounds Sterling 75,000 in respect of Programs or 125 percent of the charges for the Program that is the subject of the claim or which otherwise gives rise to the claim.

The following item is added:

3) breach of IBM's obligations implied by Section 12 of the Sale of Goods Act 1979 or Section 2 of the Supply of Goods and Services Act 1982.

The following paragraph is added at the end of this Section:

IBM's entire liability and your sole remedy, whether in contract or in tort, in respect of any default will be limited to damages.

Z125-3301-10 (10/97)

End of Document