IBM MQ for Windows - Trigger monitor service

Originally released as SupportPac MA7K

Version 1 Release 6

Last Modified: 15 September 2015

Original contributors: Wayne Schutz, Jeff Lowrey

Table of Contents

Change history

15 Sept 2015 Version 1.6.1

1. Addition of more debug output for trigsvc.exe when EventLevel > 19
2. Addition of -d flag to both setup.exe and remove.exe for more reporting
3. Addition of Comment registry key to provide information about what ini file created the service in question.

14 Nov 2014 Version 1.6.0

4. Deprecation of Notes functionality
5. Removal of Notes binaries
6. Added support for MQ v8.0
7. Added support for channel user names and channel passwords
8. Rewrote documentation rather than plain text file.
9. Take MQ Service Object functionality out of beta.
10. Addition of an update.exe program to change the registry from the console.
11. Released to GitHub

01 Jul 2011 Version 1.5.0

1. Fix for bug with GetLastError instead of errno
2. Added Support for 64bit Windows
3. Removed Support for MQ v6
4. Added Support for MQ v7
5. Removed of support for Notes API v 5.0.11 and 6.5.4
6. Added support for Notes 8.5.x
7. Added support for running MQ Service Objects on the client side (BETA)
8. Added documentation on MQ security requirements for MA7K functions

28 Aug 2006 Version 1.4.1

1. Support of SSL Key Repository specification in setup file
2. Ability to specify service Userid and Password at install time
3. Added MQRC_CONNECTION_QUIESCING to list of "retry" reason codes
4. Tested on Win/XP
5. Tested with WebSphere MQ V6
6. Linked with Lotus Notes libraries versions 6.5.4 and 7.0.1
7. Maintainer now Jeff Lowrey

16 Aug 2004 Version 1.4.0

1. Added ability to specify many of the client channel parms in the setup.ini file. See that file for details.
2. Cleaned up the path info in ma7k.mqsc sample.

13 Nov 2003 Version 1.3.2

1. Included sample mqsc file
2. Tested with WebSpere MQ v5.3.1 on Win2K
3. Added "ExitProcess" to Notes trigger to clean up "left around" DOS windows. (also, see note at bottom of this document)
4. Linked with v6.0.1 of the Notes API (A version of the module built with v5.0.11 of the library is also included, see comments below)
5. Added new retry condition
6. Updated to use new product name
7. Service name can be defined by customer which allows multiple services to run concurrently

29 Jan 2001 Version 1.3.1

1. Fixed problem with launching notes agents when ma7k is installed into a directory with blanks in the name.
2. Linked with v5.0.3 of the Notes API

07 May 2000 Version 1.3

1. Added ability to use a "Notes.ini" file when triggering Lotus Notes Agents
2. Restructured setup data file to allow specification of all parameters held in registry.

08 February 2000 Version 1.22

1. Fixed array overflow error which could cause Notes to abend if an error was reported.
2. Linked with latest Notes API toolkit.

15 December 1998 Version 1.21

1. Added MQGMO_CONVERT option on MQGET

2 November 1998 Version 1.20

1. Support for multiple threads watching multiple initiation queues.

08 August 1998 Version 1.10

1. Support for triggering Lotus Notes agents.
2. Ability to name MQM DLL to load.
3. Fixed various small problems with the format of the trigger message
4. Made specifying the .EXE on the ApplicId field optional.

07 May 1998 Initial Version

Overview

This package provides a trigger monitor (TM) which runs as a Windows service, and is intended to be used with the IBM MQ client (mqic32.dll). (The service can be run as a "local" MQ program with some restrictions.)

This program has been enhanced in version 1.5 to start and stop programs on the MQ client machine that are defined as Service objects in a MQ queue manager.

This service should be started instead of the supplied client trigger monitor (runmqtmc). This program is designed to be run only by the Windows Service Control Manager (SCM) and reports its status to the Windows Event log.

It is assumed that the user understands MQ triggering, the MQ client, Windows Services and the Windows Event log.

The following files are shipped in this package:

1. readme.odt -- This file
2. setup.exe -- Installation utility
3. update.exe – configuration management utility
4. remove.exe -- Utility to remove definitions
5. trigsvc.exe -- WebSphere MQ Client Trigger Service Program

** DO NOT attempt to run this program from the command line **

6. setup.ini -- sample setup file
7. ma7k.mqsc -- sample mqsc file for creating trigger environment.

Installation

NOTE: If you have previously installed this supportpac, you must first REMOVE that version prior to installing this version. You must run the REMOVE program from the previous installation directory.

If you have previously installed this supportpac, you must re-execute the SETUP program. This version requires additional registry entries that the SETUP program sets.

STEP 1:
ENSURE THE WEBSPHERE MQ CLIENT IS PROPERLY INSTALLED AND CONFIGURED

Ensure that the WebSphere MQ client has been installed on your client machine. If you want to run the test provided with this supportpac, you must also install the sample applications. It is important that you configure the client by either using the MQSERVER environment variable or using a client channel table. See the WebSphere MQ Clients book for details.

NOTE: If you do use the MQSERVER variable, you MUST reboot the client machine before attempting to run the supportpac. Please be sure that the MQSERVER variable is defined as a "System Variable" and not a "User Variable".

STEP 2:
INSTALL THE SUPPORTPAC FROM SUPPORTPAC ZIP FILE

After unzipping the files into a suitable directory, make the directory which you unzipped the programs into the current directory. For example, if you issued:

```
unzip ma7k.zip \supportpacs\ma7k\WebSphereMQClientTriggerService
```

then issue:

```
cd \supportpacs\ma7k\WebSphereMQClientTriggerService
```

When you are in the directory which contains the unzipped files, issue "setup" to install the service. The setup program can read the initiation queue name and queue manager name from either the command line or a configuration file.

If you wish to specify the queue name on the command line, setup program takes two optional parameters. The first optional parameter is the name of the initiation queue that the service should open and receive trigger messages from. If this parameter is not supplied, the queue defaults to: "SYSTEM.DEFAULT.INITIATION.QUEUE".

The second parameter is the queue manager name which the service should try to connect. If not supplied, it defaults to "".

If you wish to specify more than one initiation queue, or wish to have multiple threads "watching" initiation queues, or wish to change other default values, edit the supplied "setup.ini" file. (or make a copy of the setup.ini file giving it a name of your choosing) Then call setup with the -f flag:

```
setup -f configurationfile
```

For example:

```
setup -f "mysetup.ini"
```

If you want to change the name of the initiation queue or queue manager after running setup, you may do so by re-running the setup program from the install directory with the desired parameters.

See the supplied setup.ini file for the options that it supports.

INSTALL THE SUPPORTPAC FROM GITHUB

Download the repository as a zip file or clone to your desktop.

Either use the location you've downloaded, or copy the ./bin directory to a location of your choice – with a name that reflects your intended setup and site standards.

Follow the instructions for the setup as per the install from the supportpac zip file, including the optional multiple services options below.

STEP 3:
TEST THE SUPPORTPAC:

Once the SupportPac has successfully been installed, open the Windows "Control Panel" folder (found in Administrative Tools of the Control Panel folder), double click the "Services" icon. You should see a service named "WebSphere MQ Client Trigger Service". Select that line and start the service. The service should start.

The installation program defines the service as Automatic Startup, so after rebooting the client machine, the service will start automatically.

Setup a queue, process and initiation queue as any normal triggering environment. The default initiation queue is "SYSTEM.DEFAULT.INITIATION.QUEUE".

You may test the operation of the supportpac by running the sample amqsreq.exe program:

1. On the server, edit the supplied sample definitions file "ma7k.mqsc" and change to location of the MQ sample program amqsechc.exe (which is located on the client machine).
2. On the server, run: runmqsc < ma7k.mqsc
3. On the server, run the sample program: amqsreq MA7K.TEST and provide it with some messages at the command prompt.
4. If all is well, a process should start on the client machine (running amqsechc.exe) and the message you typed should be sent back. If you look at the Windows event log, you should see an entry for the trigger event.

STEP 4:
(optional) CREATE MULTIPLE SERVICES:
You may run multiple copies of this service by creating a second installation directory, unzipping the supportpac into the directory and giving the service a different name. This is most likely to be used when the customer wants to run the services under different Windows userids or further application isolation.

To do this, follow these steps:

1. Create an install directory and unzip the supportpac as in steps #1 and #2 above (except do NOT run setup yet!) **Hint**: You might want to name the directory to correspond to the service name
   (*Although this is not necessary*).
2. Use notepad (or other editor) to edit the file "servicename" in the install directory.
3. Change the servicename to one of your choosing. Blanks are allowed and up to 127 characters may be used. Be careful of special characters as Windows might not accept them as valid service names and registry key entries.
4. Modify setup.ini or other configuration files as in step #2 above. Run setup now to create the service.
5. Use the Services applet to modify the service to your choosing (changing userid / passwords, for example).
6. The "servicename" file must remain unmodified in the install directory. Each time the service starts it will read this file, so DO NOT change it.
7. To remove the service, run "remove" from the directory.
8. To create additional services, repeat the above steps. There is no built-in limit on the number of services you can define.

Using MQ Service objects:

This version of MA7K, allows you to use an MQ Service object as the model for a process that will be launched on the client machine - rather than as part of the queue manager startup on the queue manager machine.

To use this function from MA7K, you will need to specify an INI file. In that ini file, you will specify a stanza like

```
** This defines a simple service
Service:
ServiceName=SYSTEM.DEFAULT.SERVICE
```

The Service stanza otherwise supports all of the same keywords that the Thread stanza does (other than
TriggerQueueName and TriggerQueueMgrName). This would instruct MA7K to create a new thread for this Service object, run an inquiry against the queue manager to fetch the definition of SYSTEM.DEFAULT.SERVICE, and then perform the following steps:

1. execute the STARTCMD portion of the service definition with the STARTARGS as the command arguments
2. open a temporary dynamic queue and loop over an MQGET with wait for either the qmgr to quiesce, the MA7K service to shut down, or a message to appear on this queue with an MQ Feedback code of MQFB_QUIT.
3. When one of those conditions has occurred, execute the STOPCMD portion of the service definition with the STOPARGS as the command arguments.

For an additional safety factor, MA7K will do preprocessing on STARTCMD and STOPCMD to remove the literal value 'MA7K' from the front of it. This way, you can make the STARTCMD and

STOPCMD invalid as written, so that if someone actually instructs the queue manager to start or stop this Service, these commands will fail on the queue manager machine.

For example, if you have a service that has "c:\Apps\myProgram.exe" as the STARTCMD, and you want to ensure that an instanceof myProgram.exe that might exist on the queue manager machine will never get run, you can set the STARTCMD to the value "MA7Kc:\Apps\myProgram.exe". The supportPac will strip out the MA7K value and execute the original command. This is a case-sensitive check, the STARTCMD or STOPCMD *must* start with exactly 'MA7K', and not 'ma7k' or 'mA7K' etc.

Security considerations

MA7K needs the following MQ permissions granted to the user executing the Windows service in order for MA7K itself to run a thread acting as a trigger monitor.

```
+conn +inq on qmgr
+get on any trigger queues
```

Other privileges may be needed to allow any triggered applications to function.

If you have configured MA7K to run a Service object, you will also need the following additional privileges

```
+put to command queue (SYSTEM.ADMIN.COMMAND.QUEUE),
+get to model queue (SYSTEM.DEFAULT.MODEL.QUEUE)
+dsp on qmgr
+dsp on service object
```

Please note that the only messages that MA7K sends to the SYSTEM.ADMIN.COMMAND.QUEUE are PCF INQUIRE SERVICE messages, and it only inquires against the specific services you have told it to inquire against. It does not, for example, gather information on ALL services, or on queues or send any PCF messages that CHANGE or CREATE anything.

In the course of working with a Service thread, MA7K will create at least one dynamic queue based on SYSTEM.DEFAULT.MODEL queue with a name that starts with "MA7K.ADM.", and at least one dynamic queue with a name that starts with "MA7K.SVC."

MQ v8 Security considerations

There are additional security considerations if you want to present username/password credentials to an MQ v8 queue manager.

MA7K will support the use of username/password credentials over a client connection. In order to specify these, you will need to use an INI file. For the relevant stanza or global block, use a CHANNELUSERNAME field. When you run setup and specify an INI file that includes a CHANNELUSERNAME field, setup will interactively prompt you for the password for that channel user. It will not display that password on the screen as you type it. MA7K will then encrypt the password using Windows API functions (CryptProtectData) and store a hex representation of the encrypted password in the Registry.

The CryptProtectData function requires that the user who encrypts the data is also the user that decrypts the data. This means that you must run setup.exe as the user that will be running the MA7K Windows Service. This is **not** the same as the Channel User.

In order to install a Windows Service – that is, for setup.exe to complete successfully – the user running setup.exe must be a Windows Administrator. But this is only necessary when installing the service. It is not required after the service is installed.

So if you want to use MQ v8 channel credentials, these are the steps:

1. Determine the user that will run the service. (create it if needed).
2. Add that user to the Administrators Group.
3. Log in as that user
4. Unzip the MA7K package into the directory of your choice
5. Create an INI file that includes at least one CHANNELUSERNAME field

6. Run setup.exe, passing in the name of the INI file.
7. Enter the Channel User Password (s) when prompted.
8. Configure the new Windows Service to run as the service user, using the Services control panel.
9. Remove the service user from the Administrators group.

Operational Concerns

The service will write all informational, warning and error messages to the Windows Event log. You can open
"Event Viewer" from the "Administrative Tools" selection in the "Control Panel". Event messages are written to the "Application Log".

This SupportPac will not write invalid trigger messages to the dead-letter-queue. However, a warning event will be sent to the Windows Event log. If the SupportPac cannot connect to the queue manager, or the connection is broken, the SupportPac will retry the connection. By default, the following values are used for retry:

```
ShortRty = 10
ShortTmr = 60 (seconds)
LongRty = 999999999
LongTmr = 1200 (seconds)
```

The Service Applet will show the service as "Paused" while it is in retry mode. The service applet display does not automatically refresh the service's status. You must "Close" the applet and then double-click on the "Services" icon in the "Control Panel" folder to refresh the display.

The SupportPac logic will also retry if any of the following conditions are encountered:

```
MQRC_Q_MGR_NOT_AVAILABLE
MQRC_CONNECTION_BROKEN
MQRC_GET_INHIBITED
MQRC_OBJECT_CHANGED
MQRC_OBJECT_IN_USE
MQRC_Q_MGR_QUIESCING
MQRC_Q_MGR_STOPPING
MQRC_Q_MGR_NAME_ERROR
MQRC_CONNECTION_QUIESCING
```

The SupportPac will stop if too many retries have been made (more than LongRty + ShortRty). An error event is written to the Windows Event log if this happens.

The SupportPac issues an MQGET with wait against the initiation queue named. By default, the wait interval is 60,000 milliseconds. You may change this value by specifying a different "WaitInterval" value in the setup file. This is done so that unexpected terminations of the client connection do not case a "hang" condition against the initiation queue.

You may setup the channel to use Heartbeats (HBINT) if it is supported in your environment. If heartbeats are supported, you may set the MQGET wait interval to a very large value or the value -1 (which means unlimited).

When starting the service, you may temporarily override the initiation queue and queue manager name by supplying "Startup Parameters" on the service applet. If you do provide parameters, the first parameter is the queue name and the second parameter is the queue manager name. Both parameters are optional.

Altering the service parameters

All of the values you can set in the ini file can then be changed in one of two ways.

1. Running the update.exe and responding to the prompts
2. editing setup.ini file and re-executing the setup program.

After either of these steps, you must manually restart the trigger monitor. Normally, each time a triggered process is started, an event message is written to the Windows Event log. This may be changed by changing the value of the "EventLevel" key in the setup.ini file and re-executing "setup".

The following values are defined for this key:
1. All error events and start-stop events are logged
2. (default) -- error events, start-stop events, and trigger events are logged.
3. All events, including debugging events, are logged.
4. Same as level 3 except all MQGET (including MQRC_NO_MSG_AVAILABLE) messages are logged.

Removal

If you want to remove the definitions for this service, you may use the "remove.exe" command. Change into the directory where the "remove.exe" command is, and type "remove". You may now manually delete the files in the WebSphere MQ Client Trigger Monitor directory.

Running the trigger monitor as a local application

MA7K may also run as a local program, instead of as an MQ client. To run MA7K as a local program, edit the setup.ini file, changing the following keyword in the "Global" stanza:

```
MQSeriesDLL=mqm.dll
```

To run as a client, edit the key back to:

```
MQSeriesDLL=mqic32.dll
```

Common Problems

Ensure that you have looked at the Windows Event log, as this SupportPac writes all messages to that log.

To view the Event log:
1. Open "Event Viewer" from the "Administrative Tools" selection in the "Control Panel" folder.

2. Select the "Application Log" in the left hand pane.

3. The WebSphere MQ Client Trigger Monitor events are now displayed.

If you get a message box saying the service could not be started: If the service cannot start normally, you may see an error message box from the service control manager.

If you see this message box, you should look at the messages in the Windows Event log to determine the cause of the failure.

If you get: --- 2058 Return Code from MQCONN
If you are having problem connecting to the queue manager, and you are using the MQSERVER variable (instead of a client channel table), please be sure that the MQSERVER variable is defined as a "System Variable" and not a "User Variable". There is a known problem if the "applicid" has blanks in the path. This is currently due to a restriction in NT and hopefully can be fixed sometime in the future. Use the DOS shortname for the directory instead, which you can determine with "DIR /x". So, if the directory name is: "c:\spaces and blanks\executable.exe".

The output of dir /x might show:

```
10/03/2001 09:07a <DIR> SPACES~1 SPACES AND BLANKS
```

and therefore make the applicid:

```
applicid('c:\spaces~1\executable.exe')
```

Triggering Lotus Notes Agents

Deprecation of Function

As of the 1.6 release of MA7K, all of the Lotus Notes functions are deprecated. This means that no Notes compatible binaries are included in this SupportPac. All of the functions to use these binaries are still included, but no compiled files are shipped. You can still compile MA7K locally and enable the Notes functions. Instructions to do so are included at the end of this section.

Notes Triggering

This SupportPac will attempt to trigger a Lotus Notes agent if it receives a trigger message with application type of "22".

This SupportPac uses the following keys in the setup file for Notes functions.

- Tracing is controlled by the "EventLevel" key.
- The key "AgentRedirStdout" controls whether or not the output from the agent is recorded in the Windows Event log. Valid values are "Yes" or "No" (the setting is not case sensitive).
- The key "NotesIni" determines where the Notes INI configuration is read from. See below.

Edit the setup.ini file and specify a "NotesIni" key word in the "Threads" stanza(s).

If you specify a NotesIni keyword, the value may be either "Envrdata", "Userdata", "Trigdata" or the actual path and filename of the Notes ini file. If you specify "Envrdata","Userdata".etc, this means that the path and name of the Notes INI file comes from the Envrdata,Userdata, etc field of the trigger message. If you specify the name of the Notes INI file in the setup data file, that INI file is used for all invocations of the Notes agent on that Thread. See "setup.ini" file for samples.

Compiling Notes functions

The source code for triggering agents is still included with this SupportPac. The makefile and other needed utilites are also included. No binary files are included. No Notes library files or DLLs are included. If you wish to trigger Notes applications using MA7K, you must follow these instructions.

Dependencies

You must install the following three software packages on the machine you wish to use to compile the Notes functions of MA7K:

1. WebSphere MQ – at least the client install, including the SDK package. A minimum level of MQ v7.0.x is required, with 7.5 or later recommended.
2. The Notes API package for the version of Notes you wish to use.
3. A Microsoft compilation envrionment – some flavor of Visual Studio C/C++. The correct level for the Windows OS you are running is highly recommended. Nothing older than VisualStudio for XP should be used.

Setup

If you have not already done so, unzip the ma7k.zip file into a directory of your choosing.
Make a backup copy of the file “makefile” in the extracted MA7K folder.
Edit the original makefile using Notepad or any text editor to change the following items:

1. Change the value of NOTESDRIVE to point to the parent folder of your Notes API installation. (So if you have installed the Notes SDK in c:\APIs\NotesAPI, then you would set

NOTESDRIVE to C:\APIs)

2. Change the value of NOTESDIR to point to your Notes API installation. (In the above example this would be \NotesAPI)
3. Likewise, change MQDRIVE and MQDIR to point to your WebSphere MQ installation
4. Find the section of the makefile that declares LFLAGS. Remove the # from the line that includes the Notes APIs, and comment out the one that only mentions the MQ Libraries
5. Find the line that starts with “includeNotes:”. Remove the word “packageWithNotes” from the end of this line.
6. Save the file.

Compilation

Open a Visual Studio C/C++ command prompt.
Change directory into the location you have unzipped the MA7K package.
Run the following command:

```
nmake -k includeNotes
```

If there are any error messages complaining about linking problems, then re-edit the makefile to correct any file locations or file names based on NOTESDIR or MQDIR.

If you have any other compilation issues, then please do an internet search for the error message you get. Once the compilation has succeeded, you can then use the setup.exe and other binary files in this directory as normal. You can then follow the normal instructions for configuring the MA7K INI file to include Notes information.