# BlockIP2
# The Manual


March 2015
Version 3.00

Fifth Edition, March 2015

This edition applies to version 3.00 of BlockIP2 and all subsequent releases and modifications until otherwise indicated in new editions.

# BlockIP2 - The Manual

## Table of Contents


Introduction ..... 5
Avoid over-authorization by mistake ..... 6
New Features in 2.95-3.00 ..... 7
New Features in 2.94 ..... 7
Older Features ..... 7
Installation ..... 8
Windows 2008/2012 ..... 8
MQ Version 7.1 / 7.5 ..... 8
MQ Version 8.0 ..... 8
z/OS ..... 8
Linux (Intel 64 bit) ..... 10
AIX MQ 6.0/7.0 ..... 11
Sun/Solaris SPARC ..... 11
Sun/Solaris (x86_64) ..... 11
HP-UX ..... 11
Activation ..... 12
The sequence of checking ..... 13
Configuration ..... 14
A simple filtering ..... 14
Flags implemented ..... 14
Configuration file ..... 17
Configuration examples ..... 21
Enhanced SSL ..... 25
Single configuration file for multi channels/queue managers ..... 29
Specifying patterns ..... 31
Using LogFormat ..... 33
Making BlockIP2 silent ..... 33
BlockIP2 and Firewalls ..... 34
Filtering sequence ..... 35
Troubleshooting ..... 35
Log file example ..... 36
Compilation ..... 37
AIX and MQ version 7.1 / 7.5 / 8.0 ..... 37
Solaris_x86 and MQ version 7.1 / 7.5 / 8.0 ..... 37
Solaris SPARC and MQ version 7.1 / 7.5 / 8.0 using gcc ..... 37
Solaris SPARC and MQ version 7.1 / 7.5 / 8.0 using cc ..... 38
Linux86_x64 and MQ version 7.1 / 7.5 / 8.0 ..... 38
Linux86 and MQ version 7.1 / 7.5 / 8.0 ..... 38
HP-UX and MQ version 7.0 ..... 39
z/os and MQ version 7.0.1 / 7.1 / 8.0 ..... 39
z/Linux and MQ version 7.1 / 7.5 / 8.0 ..... 39
Windows and MQ version 7.0 / 7.5 / 8.0 ..... 39

OS/400 ..... 39
Messages ..... 40
Considered changes ..... 43
Trademarks ..... 43
End User Agreement ..... 44
Change History ..... 45

# Introduction

It has always been a challenge to keep our IBM MQ Networks secure. Today in MQ IBM offers us the ability to use SSL to prevent intruders. Anyway SSL requires some administration and creating an infrastructure that supports administration of SSL, CA etc.

When I started working with IBM MQ (MQSeries 1.2) we used it only as an internal tool, and there was no client connections allowed. That was a manageable solution. After the introduction of MQ-clusters and the usage of Client Connections exploded it became a challenge to keep the IBM MQ network secure.

This was the trigger to start building a simple IBM MQ Security exit that was able to help me keep our networks secure, I had to do a lot studying the intercommunication guide; I did a password verification exit, similar to CSQ4BCX3 (supplied together with MQ 5.3.1 for z/OS). When this exit was finished I began to look on the security, and saw that we also had to do some IP-filtering, and I needed the ability to log connection attempts, and sometime had to do investigation when spammers entered the network, and was trying to connect, using a channel. This lead to development of the LogIP and BlockIP exits. The z/OS assembly exit version of LogIP and BlockIP are kept active to help z/OS system programmers in case they need a model for developing a new exit for special purpose. The two exits are still downloaded (2015). Not with the same frequency as BlockIP2.

When the exits was released late 2002, I could see that it was an success, I had some meetings on the T&M conference in Las Vegas in 2003, and I could feel that there was a need to promote the exit, because it was the same situation all WebSphere MQ administrators was facing: Keep MQ tight.

What started just as a simple exit has now been ported to z/OS, OS/400, Linux, Windows, HP-UX, and Sun/Solaris. A lot of specialists round the globe have contributed to BlockIP2 with requirements, coding, testing etc.

The MAXCHL connection control was originally based on the SupportPac ME71 written by Tony Madden, MQWare UK. Under z/OS, Windows and UNIX/Linux I've moved into shared memory, to gain high performance without a high CPU consumption. This functionality is now implemented by IBM in WebSphere MQ version 7.0.

Duke Nguyen has extended BlockIP2 with support for integration with Object Authority Manager (OAM) for working authentication. Authentication is NOT done by BlockIP2 but will require an OAM authorization service. Additional logic has been implemented in 2.93-2.94 to support bypass of OAM if the client application request it.

This was originally described by Ben Ritchie from IBM on developerWorks®:
**Enhancing WebSphere MQ JMS security using the Object Authority Manager**
http://www.ibm.com/developerworks/websphere/library/techarticles/0512_ritchie/0512_ritchie.html.

This solution is purely based on a filtering solution after WebSphere MQ version 7.1 was released it was my plan to sunset the development, however there are still room for features.

The latest implementation of FAP control was implemented to trap old WebSphere MQ Client applications that needed an upgrade so they were using a supported foundation. Prior to this extension the only way to trap them was either issue a software scan or running a trace. Both approaches are time/resource consuming.

Here in 2015 there is still a demand for the BlockIP2 security solution even with the latest security enhancement built into IBM MQ version 8.0 with CHLAUTH. This is mainly due to the ease of use and understands how the configuration works and the simple deployment in large organizations.

## Avoid over-authorization by mistake

Over authorization is where you actually have given too much authority/access, no one complaint. Those who have it are not complaining. Maybe they don't know. Then they are in good faith. No one except the MQ administrator is to blame.

If you have this rule in the BlockIP2.ini configuration file on a UNIX/Linux queue manager:

```
CON=*;*;MCA=mqm;
```

You actually grant anyone connecting to the queue manager administrative rights, was that the meaning?

**No way!**

Administrative authority should only be given when proper authentication has been conducted so you know who is in the other end. This can be achieved using SSL/TSL.

## New Features in 2.95-3.00

- Support for MQ 8.0 on windows (64 bit exit)
- Support for Partner name check to control name of remote queue manager.
- Support for FAP (MQ Client connection software level) check to discontinue connections from old MQ clients if needed.

## New Features in 2.94

- The handling of USESECPARMS has been enhanced to allow it to work without defined users, so the passed information can be bypassed. Either to allow the usage of MCAUSER on the channel or from previous CON=/SSL= statements.

## Older Features

- Support for Multi instance queue managers added
- Z/OS configuration caching, thanks to Paul Giordano
- Moved shared memory anchors from mqm folders to /tmp, to stay clear of changes in future MQ releases.

# Installation

When you have downloaded the BlockIP2.zip and unpacked the files, it's time to install the Exit.

## Windows 2008/2012

### MQ Version 7.1 / 7.5

Just copy the BlockIP2.dll and BlockIP2S.exe into the "exits" directory where IBM MQ is installed, this makes it very easy to configure the exit. It's VERY important that BlockIP2S.exe is placed in same directory as BlockIP2.dll. This program is used to establish shared memory for BlockIP2.

IBM MQ version 7.1 / 7.5 for Windows still supports the 32 bit exits, so no problems here.

### MQ Version 8.0

Just copy the BlockIP2.dll and BlockIP2S.exe into the "exits64" directory where IBM MQ is installed, this makes it very easy to configure the exit. It's VERY important that BlockIP2S.exe is placed in same directory as BlockIP2.dll. This program is used to establish shared memory for BlockIP2.

IBM MQ version 8.0 requires a 64 bit version of the exit.

**NOTE: The default log path is changed, so you will now find it in where WebSphere MQ loads the exit. The default BlockIP2.ini is also loaded using this path.**

## z/OS

You have to upload blockip2.load.mvs and maybe blockip2.obj.mvs and blockip2.c.mvs (containing BlockIP2.c), blockip2.jcl.txt.

I have added the BlockIP2.c in Xmit format ready to upload, without having to deal with the ASCII to EBCDIC conversion; you just upload it to z/OS using this ftp command:

**ftp -s:upload.txt hostname** (Just remember to change <userid> and <password> to yours before using it, or use another way to upload the file binary to a PS with a fixed record length of 80).

```
<userid>
<password>

bin
quote SITE recfm=fb lrecl=80 blksize=8000
put blockip2.c.mvs
put blockip2.load.mvs
put blockip2.obj.mvs
ascii
quit
```

Example upload.txt

When you have uploaded **blockip2.LOAD.MVS, blockip2.c.mvs** and **blockip2.obj.mvs** you have to issue **receive** to get into a usable dataset:

**TSO RECEIVE INDATASET(BLOCKIP2.C.MVS)**
**TSO RECEIVE INDATASET(BLOCKIP2.LOAD.MVS)**
**TSO RECEIVE INDATASET(BLOCKIP2.OBJ.MVS)**

This will give you three PDS with different formats.

When you successfully have uploaded the stuff, it's time to decide what to use. The delivered load module, the object or compile it your self.

BLOCKIP2.LOAD.MVS contains a ready to use version of the security exit.

Assemble and compile but before you're able to do that you have to change the JCL to match your installation.
**You can use the supplied JCL in BLOCKIP2.MVS BUILDJCL for version 7.0 / 7.1** and just change the libraries to you installation standard. This means that you don't need to compile the programs just link them.

If you like to compile the whole stuff: first submit blocki2w in BLOCKIP2.OBJ.MVS, to create the WTO routine. When this is done submit blockip2.jcl (please check the link step, should contain the same includes as in BULDJCL). When this is run ok, you should have a working security exit.

On your xxxxCHIN task add a //SYSPRINT DD card to catch output from BlockIP2, and you can add a DD card to handle the configuration file. The link-edited modules must be placed in the data set specified by the CSQXLIB DD statement of the channel initiator address space procedure; the names of the load modules are specified as the exit names in the channel definition.

The exits are invoked as if by an z/OS LINK, in:

- Non-authorized problem program state
- Primary address space control mode
- Non-cross-memory mode
- Non-access register mode
- 31-bit addressing mode

You can however upload BlockIP2.c to z/OS using FTP and handle ASCII to EBCDIC conversion like this (just remember to create a dataset named 'MQ.BLOCKIP.C, with the LRECL=255 and RECFM=VB):

```
<userid>
<password>


cd 'MQ.BLOCKIP.C'
quote SITE SBDataconn=(IBM-1047,ISO8859-1)
put blockip2.c
quit
```

myftpparm.txt

**ftp -s:myftpparm.txt hostname**

The trick is done by **`quote SITE SBDataconn=(IBM-1047,ISO8859-1)`** which tells z/OS to do the conversion. The codepages IBM-1047 and ISO8859-1 differs depending on where you are located, so talk to your z/OS Unix Systems Service administrator to obtain the correct values in your situation. This conversion can also be used when you download a coded file from z/OS.

## Linux (Intel 64 bit)

Just untar BLOCKIP2.TAR from Linux_x86_64 subdir in /var/mqm/exits64, and you're almost ready to go.

You will need to change the owner using root authority:

```
chown mqm:mqm /var/mqm/exits64/BlockIP2
chmod 550 /var/mqm/exits64/BlockIP2
```

## AIX MQ 6.0/7.0

You have to compile the exit yourself, because I don't have this platform.

Compilation is documented on page 37

## Sun/Solaris SPARC

You have to compile the exit yourself, because I don't have this platform.

Compilation is documented on page 37

## Sun/Solaris (x86_64)

You have to compile the exit yourself, because I don't have this platform.

Compilation is documented on page 37

## HP-UX

You have to compile the exit yourself, because I don't have this platform.

Compilation is documented on page 37

## Activation

You specify that WebSphere MQ have to use the security exit by using the SCYEXIT and SCYDATA keywords when defining/altering channels. This could be like this:

```
ALTER CHANNEL(MQT2.TCP.MQT1) chltype(SVRCONN) +
   SCYDATA('172.20.10.*;172.221.*') +
   scyexit('BlockIP2(BlockExit)')
```

This works on Linux, Windows.

If it was on z/OS it would look like this:

```
ALTER CHANNEL(MQT2.TCP.MQT1) chltype(SVRCONN) +
   SCYDATA('172.20.10.*;172.221.*') +
   scyexit('BLOCKIP2')
```

For further information on the commands refer to the MASC. and Intercommunication guides.

## The sequence of checking

The checking is done in the following sequence:

1. CheckConnectionPattern()
   Basic checking of connection names, based on Patterns= keyword

2. CheckBlockedUserId()
   Basic checking of banned userids, based on BlockUsers= keyword

3. CheckUserId()
   Basic checking of authorized userids, based on Userids= keyword

4. CheckPartnerName()
   Check actual name of partner queue manager or logged on userid, based on PartnerName= keyword.

5. CheckFAP()
   Check software level of partner.

6. CheckCONList()
   Advanced connection checking, based on CON= keywords

7. CheckSSLList()
   Advanced connection checking, based on SSL= keywords

8. CheckBlankUser()
   Checking for blank userids, controlled by +b option and AllowBlankUserID= keyword.

9. CheckInvalidUsers()
   Checking for invalid userids like mqm, MUSER_MQADMIN, controlled by –n option and BlockMqmUsers= keyword.

This sequence should help to explain how BlockIP2 behaves.

When developing configuration files, it's recommended to use either basic debug mode '–d' or technical debug mode '-d1'. This will give you valuable information about what BlockIP2 are doing. This will help you to create a solid shield to protect your MQ infrastructure.

# Configuration

As you saw in the introduction, you control the behavior of BlockIP2 using SCYDATA().

You can specify a simple pattern, or use keywords.

The syntax is:

SCYDATA('[NONE;|{pattern[;pattern[;pattern...]]][;userids][;flags]}|FN=filename;[flags;]}')

SCYDATA() acts as SCYDATA('NONE;') to ease configuration.

## A simple filtering

SCYDATA('172.20.*') to allow anybody from the network 172.20.anything to access the queue manager.

Yet another SCYDATA('172.??.10.21') this one allows only connections from 172.10.10.21 172.11.10.21, it requires two digits in second IP byte.

SCYDATA('NONE') is used together with the default BlockIP2.ini configuration file to allow specification without defaults.

## Flags implemented

| Flag | Description |
|---|---|
| FN= | Name and path to configuration file. |
| NONE; | indicate that no additional patters are added when using default configuration file BlockIP2.ini |
| -d | Debugging activated |
| -d0 | Special for debugging of specification file |
| -d1 | Technical debugging |
| -d8 | Development debugging |
| -i | Ignore default configuration file BlockIP2.ini |

| | |
|---|---|
| -l | Switch off recycle of logfiles |
| -z | z/OS only: Translate [] to <> on messages |
| -n | Block system accounts: mqm, musr_mqadmin and "" (unspecified)* |
| +b | Allow blank userids (unspecified) to connect. (added in version 2.21)* |
| -w | Use event log on Windows or syslog on UNIX |
| -m | Active configuration caching** |
| -s | Allow self signed certificates |
| -t | Disable new shared memory anchor points |
| -q | quiet mode, suppress connect and disconnect messages from BlockIP2 logging. |

*) **NOTE:** -n and +b options is only accepted if FN= mode is note used. Use AllowBlankUserID=Y; and/or BlockMqmUsers=Y; in configuration instead.

**) **NOTE:** -m option is z/OS specific.

To enable the debugging you specify:

SCYDATA('172.20.*;**-d;**')

Using a specification file (the way of specifying the file differs from platform to platform):

**On windows:**
```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
  SCYDATA('FN=c:\path\Blockspec.txt;') +
  scyexit('BlockIP2(BlockExit)')
```

**On Linux:**
```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
  SCYDATA('FN=/var/mqm/exits/Blockspec.txt;') +
  scyexit('BlockIP2(BlockExit)')
```

**On AIX:**
```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
  SCYDATA('FN=/var/mqm/exits/Blockspec.txt;') +
  scyexit('BlockIP2(BlockExit)')
```

**On HP-UX:**

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
  SCYDATA('FN=/var/mqm/exits/Blockspec.txt;') +
  scyexit('BLOCKIP2')
```

**On SUN/Solaris x86:**

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
  SCYDATA('FN=/var/mqm/exits/Blockspec.txt;') +
  scyexit('BlockIP2(BlockExit)')
```

**On z/OS:**

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
  SCYDATA('FN=//DD:BLOCKDD;') +
  scyexit('BLOCKIP2')
```

Or if you're using a PDS, you can specify the member name as well:

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
  SCYDATA('FN=//DD:BLOCKDD(MEM);') +
  scyexit('BLOCKIP2')
```

Or using a USS file placed in HFS:

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
  SCYDATA('FN=/var/mqm/exits/Blockspec;') +
  scyexit('BLOCKIP2')
```

Now what can we specify in the configuration file??

## Configuration file

Following keywords are implemented:

**Please note that all keywords are written on a new line, it's not allowed to specify more keywords on the same line! Only the first one is considered as a command, the rest is accepted as a comment.**

| Keywords | Description |
|---|---|
| AllowBlankUserID=Y; | Allow blank userids to connect. **To Enable this option can lead to security risks.** |
| AllowSelfSignedCertificates={Y\|N}; | Allow the use of self-signed certificates. This is by default blocked. |
| ASC={Y\|N}; | Allow the use of self-signed certificates. This is by default blocked. Short form of "AllowSelfSignedCertificates=" |
| BlockMqmUsers={Y\|N}; | Flag to block for system accounts: **mqm, musr_mqadmin** and **''** |
| BlockUsers=<generic_user_list>; | List of negative userids using generic pattern matching |
| CHANNEL=<generic channelname>; | Working with QMGR= to control a combined configuration file. This is used in BlockIP2.ini |
| CON=<conname>;BLANK_USERID\|<userids> [;{MCA={*\|userid}\|BLOCK}]; | Allow manipulation/selection on specified connectionname/userid. This have higher precedence than userid/pattern filtering. BLANK_USERID is added to support some special WAS situations. |
| INCQMGR={Y\|N}; | Include short queue manager name in the log file. To make it easy to distinguish log entries from various queue managers on same system. **(Windows, AIX, Linux and Solaris only)** |
| LogCount=<nn>; | Number of generations of log file. Default is 3 and max is 99. **(Windows, AIX, Linux and Solaris only)** |
| LOGCYCLE={N\|Y}; | Control for logfile switching , default is on |
| LogDrive=<spec>; | The drive there the log file is placed **(Windows only)** |

| | |
|---|---|
| LogExt=<spec>; | extension of log file, default is 'txt' |
| LogFileName=<spec>; | Name of the log file default is 'BlockIP2' |
| LogFormat=<spec>; | M – For queue manager name only or<br>N - Name only or<br>ND - Name and Date or<br>NDC - Name Date Channel format<br>MNCD – Qmgr Name Channel Date format or any variation |
| LogPath=<spec>; | The path where the log file is placed. Default is /var/mqm/exits for UNIX. and <install path>\exits for windows. |
| LogSize=<nnnnnnnn>; | Maximum size of log file before switching to next. default is 200KB, min is 100KB. **(Windows, AIX, Linux and Solaris only)** |
| MAXCHL=<generic_ChannelName>;<br><numberOfChannels>; | Limit the number of connection to this channel, the number of connections are checked in the INIT phase. |
| PartnerName=<generic_partnername_list>; | List of allowed queue managers or logged on users. |
| Patterns=<generic_connection_list>; | List of positive network connection names using generic pattern matching<br><br>Patterns starting with a-z are treated as a DNS entry, and pattern entries starting with 0-9 are treated as a IP-address. |
| QMGR=<generic queue manager>; | Working with CHANNEL= to control a combined configuration file. This is used in BlockIP2.ini |
| QMGR_SHORT_NAME=<short_name>; | Short queue manager name to include in the log file. Up to 8 characters are accepted.<br>By default it takes the first 8 characters from the queue manager name. |
| SECPARMSOFF={Y\|N}; | Enabling this will for OAM checks off, and set LongMCAUSER is set to the userid controlled by CON=/SSL=.<br><br>**NOTE:** This can lead to security exposures if not used correctly. |
| SSL=[C=<GN>,][L=<GN>,][O=<GN>,][PC=<GN>,]<br>[ST=<GN>,][OU=<GN>,]CN=<GN>;...<br><br>[{BLOCK\|MCA={*\|userid\|xx(from,len)};] | fine selection based on the DN, which also allow override of MCAUSERID and even to blacklist one or more Certificates. You can use any of these filters in any combination. |

| | |
|---|---|
| | **The only restriction is: Imbedded blanks are not allowed.** Or have a look in the code for more information.<br><br>MCA=xx(from,len) is MCAUSER extractor from SSL DN, works like substr and start in pos 1. |
| SysLog={FILE\|ONLY\|BOTH}; | Control where output from BlockIP2 are sent. **(UNIX ONLY)** |
| SYSLOGFCLTY={LOG_DAEMON\|LOG_USER} | Control which message file BlockIP2 uses for reporting. **(UNIX ONLY)** |
| SYSLOGPRTY={LOG_INFO\|LOG_ERR\|LOG_NOTICE\|LOG_WARNING}; | Control the "priority" of BlockIP2 messages. **(UNIX ONLY)** |
| TMPPATH={Yes\|No}; | Disable new shared memory anchor path. |
| TERM={Y\|N}; | Controlling print of a termination message, including the connection name. |
| Userids=<generic_user_list>; | List of positive userids using generic pattern matching |
| UseridUpperLowerCase=*; | Swift off case sensitivity, fold any matching to uppercase. |
| USESECPARMS={Y\|N}; | Setting MCAUSER based on content passed in SecurityParms.<br>LongMCAUSER is set to the received userid.<br><br>This is designed to work with a OAM exit.<br><br>**NOTE:** If not used with a OAM exit this can lead to security exposures. |
| WinLog={FILE\|ONLY\|BOTH}; | Control where output from BlockIP2 are sent. **(Windows ONLY)** |
| WTOPFX=<prefix>; | Prefix for WTO messages, default is BLOCKIP-. Max length 20 characters **(z/OS only)** |
| USESECPARMS={Y\|N}; | Integration with Object Authority Manager (OAM) to utilize authentication.<br>LongMCAUSER is set to the received userid. |

Patterns=, Users= and BlockUsers= offers concatenation functionality.

This allows file content like:
Patterns=1.2.3.4,1.2.3.5,1.2.3.6; /* hosts 4,5,6 in the 1.2.3 network */
Patterns=1.2.1.4,1.2.1.5,1.2.1.6; /* hosts 4,5,6 in the 1.2.1 network */

and any of the six addresses will be allowed. this can also be specified as:
Patterns=1.2.3.[4-6],1.2.1.[4-6];

Patterns=mrmqdk01.mrmq.dk,mrmqdk02.mrmq.dk,spyder,10.31.*;
Will cause BlockIP2 to look the 3 hostnames up and use the returned IP address in matching the connection name. You **cannot** use a generic hostname like *.mrmq.dk

Specifying pattern rules are described in details on page 31

**There is limit on 4096 characters maximum length on Patterns, Users, BlockUsers.**

There are room for 256 SSL patterns and 64 CON patterns. This limits can easily elevated, but it require more memory. The memory obtained by BlockIP2 remains allocated until the connection is terminated, therefore have I decided to set some limitation on.

## Configuration examples

To use multipattern: just separate the patterns with a semicolon(;) like this:

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
    SCYDATA('172.20.109.*;172.221.*;10.31.*') +
    scyexit('BlockIP2(BlockExit)') *
```

**This will allow communication from any computer in the 172.20.109.*, 172.221.* and 10.31.* networks.**

You can also use a single position placeholder in the pattern:

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
   SCYDATA('192.168.??.20;10.31.*') +
   scyexit('BlockIP2(BlockExit)')
```

**This will allow any IP-address matching 192.168.10.20, 192.168.11.20.. 192.168.99.20 and 10.31.* to pass verification.**

---

How to block bad userids (mqm, musr_mqadmin and ''), this is done using the switch **-n** in SCYDATA. When the RemoteUserIdentifier of an incoming connection request contains one of the listed users above BlockIP2 sends MQXCC_SUPPRESS_FUNCTION so the communication attempt is abandoned.

BlockIP2 have been updated in SCYDATA like this:

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
  SCYDATA('172.20.10.*;172.221.*;10.31.*;-n') +
  scyexit('BlockIP2(BlockExit)')
```

---

A allow blank userid (unknown) to connect, this is done with the **+b** option.

```
alt chl(JMSCHL_SPY) chltype(SVRCONN) +
   SCYDATA('172.20.10.*;172.221.*;+b') +
   scyexit('BlockIP2(BlockExit)')
```

> ***WARNING: Using this option is causing a security hole, so anybody using a java application or a security client exit to connect to the queue manager and get up to mqm authority.***

A quiet mode option is added, so there will be only one line per connection attempt in the log. This is done with the **-q** option.

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
   SCYDATA('172.20.10.*;172.221.*;-q') +
   scyexit('BlockIP2(BlockExit)')
```

---

A debug option is added, to allow a more comprehensive logging of the activity inside BlockIP2. This done with the **-d** option.

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
   SCYDATA('172.20.10.*;172.221.*;-d') +
   scyexit('BlockIP2(BlockExit)')
```

---

BlockIP2 is able to obtain the security specification from a file. This feature was added to circumvent the 32 byte limitation in SCYDATA, and was requested from complex installations.
The filename is specified in SCYDATA like this on **windows**:

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
  SCYDATA('FN=c:\path..\Blockspec.txt;') +
  scyexit('BlockIP2(BlockExit)')
```

or like this on **Linux** and the rest of **Unix** implementations:

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
  SCYDATA('FN=/var/mqm/exits/Blockspec.txt;') +
  scyexit('BlockIP2(BlockExit)')
```

or like this on **z/OS**:

```
alt chl(MQT2.TCP.MQT1) chltype(SVRCONN) +
  SCYDATA('FN=DD:BLOCKDD;') +
  scyexit('BLOCKIP2')
```

**NB**: The **FN=**<filename>**;** requires the ending semicolon(;).

---

Here is a simple commented specification

```
# This is a comment BlockIP2 security specification Blockspec.txt
```

```
Patterns=10.1.*,172.20.31.*,127.0.?.1;
Userids=xxx,yyy,zzz*,etc,mrmq,root,us???mq;
BlockMqmUsers=Y;
#-------------------------------------------------------------
# Description of security specification:
# This specification will allow:
#  - connections from 10.1.*, 172.20.31, and 127.0.0.1, 127.0.1.1, ... 127.0.9.1
#    Networks.  Entries are separated using comma(,)
#  - allow  userids: xxx, yyy, etc, mqm and mrmq.  Together with users
# starting  with zzz<something> and users beginning with us<three chars>mq.
#
# - BlockMqmUsers=Y; means that following users are blocked: mqm,
#    MUSR_MQADMIN and musr_mqadmin.
#  All specs must end with a semicolon ; anything after ; is parsed as a comment
# Important:
# If you specify Patterns twice they are concatenated, so they are all used
# for matching.
#
# It's very important that all specs are terminated with semicolon(;) otherwise
# you will receive connection refused. Because I think it's best only to use
# positive error-free identification
# The patterns are separated using comma(,) This is also important to remember
```

BlockIP2 is able to change MCAUSER depending on the incoming connection credentials.

```
# This is a comment BlockIP34.txt
Patterns=10.1.*,10.1.11.*,user1.mrmq.dk,127.0.?.1;
SSL=CN=ibmwebspheremqclient2;MCA=mrmq; ok userid
SSL=CN=ibmwebspheremqQM2;BLOCK; blocked user
SSL=CN=ibmwebspheremq*;MCA=*; just do nothing
SSL=CN=*;MCA=NoBody; Set all other to NoBody
# EOF
```

The connection may come from: 10.1.*,10.1.11.*,**user1.mrmq.dk**,127.0.?.1 (Yes the user1 one is found using the DNS). This means that it supports the use of DHCP....

In the example above will connections with a CN=ibmwebspheremqclient2 get the MCAUSER=mrmq.

Connections from CN=ibmwebspheremqQM2 will be refused all other connections from CN=ibmwebspheremq* will use the normal userid (or MCAUSER if specified on the channel).

Connections made with CN=<anything> will have the MCAUSER=NoBody, which should be no defined. (Could also be **SSL=CN=*;BLOCK**; have the same function when user NoBody is undefined)

## Enhanced SSL

The newest enhancement in BlockIP2 is the enhanced MCAUSER extraction feature that allows BlockIP2 to assign a part of the partners DN to the MCAUSER. This gives you the flexibility to have many users sharing the same SSL= filtering schema, and the assign the authority based on their own certificates. imbedded blanks in the names are removed using truncation, meaning that a name like 'h c andersen' will be 'hcandersen', please note that the current implementation are doing the truncation after the substring. This might get changed in the future. Have a look here: Considered changes in the near future.

Syntax is:
**SSL=O=mrmq,MCA=CN(1,8);** will extract the first eight characters from CN and use as MCAUSER. The MCA=xx(from,len) recognizes the used prefixes in the SSL certificate.

Let's see the result:

```
# This example should describe the "substring" like functionality in the SSL/MCAUSER
logic.
Patterns=10.1.*;
SSL=O=mrmq;MCA=CN(1,8);   set mcauser to the Common name pos 1 thru 8.
SSL=O=acme;MCA=L(1,4);    set MCAUSER to location pos 1 thru 4.
SSL=CN=*;BLOCK;           Block the rest
# EOF
```

Here is a couple of DN, and with the rules above it should illustrate the way it work.

DN=CN=jones carin,O=mrmq,C=dk,L=copenhagen  gives MCAUSER=jonesca
DN=CN=smith john,O=mrmq,C=dk,L=esbjerg  gives MCAUSER=smithjo

DN=CN=bond james,O=acme,L=houston,C=US gives MCAUSER=hous
DN=CN=johnes john,O=acme,L=berlin,C=DE gives MCAUSER=berl

And certificates from anyone else will get refused by the SSL=CN=*;BLOCK;

---

Here is an commented example using **CON=** (not from the real world), but still informative:

```
#
# Simple filter implemented in BlockIP2 version 2.20
```

```
#
# 1. stop all connection attempts from mqm (NoBody is an undefined or blocked
#     userid).
CON=*;mqm;MCA=NoBody;
#
# 2. Stop users starting with dk14 from 10.31.* Might be a foreign network
CON=10.31.*;dk14*;MCA=NoBody;
#
# 3. Allow master03 when coming from 172.20.10.31, using the MCAUSER settings
CON=172.20.10.31;master03;
#
# 4. Allow spider when coming from 10.*, and set MCAUSER to master04
CON=10.*;spider;MCA=master04;
#
# 5. Allow us* userids when coming from 172.31.*, and set MCAUSER to the presented
#    userid eg. us12345
CON=172.31.*;us*;MCA=*;
#
# 6. Block all other attempts.
CON=*;*;MCA=NoBody;
#
# Control the number of connections:
#
# Establish a default max connection count of 5
MAXCHL=*;5;
#
# Specific channel limitations
MAXCHL=SYSTEM.DEF.SVRCONN;25;
MAXCHL=SYSTEM.ADMIN.SVRCONN;25;
MAXCHL=SYSTEM.AUTO.SVRCONN;25;
```

The list is searched from top to bottom, and when BlockIP2 detects a match on connection-id and userid, this means that it's important to specify the rules in the right order.

**Caution:** Using both SSL and CON keywords in the same specification file, should be done with great caution, because the SSL have precedence over CON. You might see that a MCA change caused by CON, can affect a SSL setting without MCA settings.

---

Making use of hostnames:

BlockIP2 version 2.55 support DNS lookup of hostnames, this makes it easier to control patterns and also support filtering of users using dynamic TCP/IP addresses.

```
#
#Allow connection from the Boss, 10.3.* and anything from 11.* ending on ".12"
Patterns=theBoss,10.3.*,11.*.12;
#
# Grant userid mrmq connecting from the Boss the MCA user mqm:
#
CON=theBoss;mrmq;MCA=mqm;
#
# Grant anybody else access using MCA=user:
#
CON=*;*;MCA=user;
#
#
```

Here is an commented example using **MAXCHL=** (not from the real world), but still informative:

```
#
# Control the number of connections:
#
# Establish a default max connection count of 5
MAXCHL=*;5;
#
# Specific channel limitations
MAXCHL=SYSTEM.DEF.SVRCONN;25;                Allow 25 connections here
MAXCHL=SYSTEM.ADMIN.SVRCONN;25;               Allow 25 connections here
MAXCHL=SYSTEM.AUTO.SVRCONN;25;               Allow 25 connections here
```

This feature is implemented using shared memory.

The MAXCHL= table is processed sequential, and all matches are processed, this means that the last match have higher priority than the first one. Therefore should you place the default value as the first entry, and place specific entries after it.

**BlockIP2 rely on /tmp/.$QMGR_BlockIP2.1 and /tmp/.$QMGR_BlockIP2.2 for pointing out the shared memory segment in the UNIX implementation. The determination follows mqs.ini/qm.ini.**

This can give a problem when working with WebSphere MQ version 7.0.1 and using the failover feature, because the allocated storage will survive.

## Single configuration file for multi channels/queue managers

With BlockIP2 version 2.60 and higher are you able to use a single configuration file to control any combination of queue managers and channels. The only limitations are simplicity. My recommendation is: **Keep it simple** because this makes you have control over the configuration and makes it easy to maintain.

An example: This file is default loaded from the queue managers exit path. The file is named BlockIP2.ini and is case sensitive for some implementations.

```
# Simple filter implemented in BlockIP2 version 2.60
# Copyright (c) 2007 M-Invent, All rights reserved.
#
#
# First some global settings:
QMGR=*;
CHANNEL=*;
#
BlockMqmUsers=Y;
#
# Specific channel limitations
MAXCHL=SYSTEM.DEF.SVRCONN;25;
MAXCHL=SYSTEM.ADMIN.SVRCONN;25;
MAXCHL=SYSTEM.AUTO.SVRCONN;25;
#
# 1. stop all connection attempts from mqm (NoBody is an undefined or blocked
userid).
CON=*;mqm;MCA=NoBody;   Could also be: CON=*;mqm;BLOCK;
#
# Now we will play with the SATURN queue manager and the system channels
QMGR=SATURN;
CHANNEL=SYSTEM.*;
# 2. We don't like peter and we have frank as administrator from 10.31.*
BlockUsers=peter;
CON=10.31.*;frank;MCA=mqm;
#
# we are still playing with the SATURN queue manager but now it's SATURN.INPUT*
channels
CHANNEL=SATURN.INPUT*;
# 3. Allow communication from 10.31.14.11 and  10.32.14.11 with user dku12345
CON=10.3[1-2].14.11;dku12345;
#
```

```
# Now we will play with the MARS queue manager and the system channels
QMGR=MARS;
CHANNEL=SYSTEM.*;
# 4. We don't like deu98765 and we have "frank" as administrator from 10.31.*
BlockUsers=deu98765;
CON=10.31.*;frank;MCA=mqm;
#
# Now we will play with the VENUS queue manager and the system channels
QMGR=VENUS;
CHANNEL=SYSTEM.*;
# 5. We have "donna" as administrator from 172.20.14.11
CON=172.20.14.11;donna;MCA=donnaadmin;
#
# Close the last hole in case a channel is unprotected. This is the last check...
QMGR=*;
CHANNEL=*;
# 6. Block any non accepted connection attempts.
CON=*;*;BLOCK;
#
# Please remember that the configuration file is processed from the top and down.
# The CON/SSL check is stopped on first match; this means that placing the stage 6.
on top
#  would block any connect attempt.. And by placing it last it last it will block
# connect attempts on non-protected channels.
#
#
```

The default locations are:
**UNIX/Linux/Solaris/HP-UX**:
`/var/mqm/exits/BlockIP2.ini`

**Windows**:
The location where BlockIP2.DLL loaded from. For example:

```
c:\program files\IBM\Websphere MQ\Exits64\BlockIP2.ini
```

The file location is found based on mqs.ini/qm.ini or registry settings, to allow more default configuration files on server hosting more queue managers.

## Specifying patterns

| Special character | Rule description |
|---|---|
| * | Any number of characters towards end of spec |
| ? | Any character in this position |
| # | Any numeric (0-9) character in this position |
| $ | Any alphabetic character a-z and A-Z |
| (0-4) (c-j) (A-z) | Any alphanumeric character in the specified range (Added for European z/os systems) |
| [0-4] [c-j] [A-z] | Any alphanumeric character in the specified range |

**Remark:** The range selection is based on the character value, this means that there are a huge difference between z/OS, AS/400 and the other platforms because z/OS and AS/400 are using EBCDIC and the rest are using ASCII, this means that [0-z] will allow any character on Linux, but is a bad specification on z/OS. There are currently (v.2.21) added a z/OS limitation that are blocking for mixing numbers and letters in a range spec.

Here are some examples of how to specify various patterns:

```
#
# Simple filter implemented in BlockIP2 version 2.21
#
# 1. using range in connection name
# This statement allow connections from connection names that are starting with 10.1 having 2-5 in pos 5, having 0-9 in 7'th
# position and anything in the last group (0-255). This also means that you can't get in from 10.12.10.11 ...
# Net we're going to look on the userid, it must start with the 'us' and 6 digits, like us0234237, and you'll not be able to
# connect if you user id us02345 (one digit too short.
CON=10.1[2-5].#.*;us######;MCA=american;
#
# 2. using alphabetic filter in userid
# This statement allows connections from 172.20.10.21 only, But the userid have to start with the letters 'gb' followed by
#  three letters and ending with two zeroes '00'. This means that the userids would
```

```
look like: gbabb00, gbwho00.
# And gbilo44 is not allowed.
CON=172.20.10.21;gb$$$00;MCA=uksys;
#
# 3. using range in userid
# This statement allows connections from 172.20.*, But the userid have to start with the letter 'd' followed by a letter
# in the range from 'a' thru 'd' followed by '00' and a letter in the range 'w' thru 'z'. This means that the userids would look
# like: da00w, dd00z. And de00a is not allowed.
CON=172.20.*;d(a-d)00(w-z);MCA=agent007;
#
# 4. using a host name in the filter
CON=server1.mrmq.dk;mrmq;
#
# 5. Block all other attempts.
CON=*;*;BLOCK;
#
```

## Using LogFormat

LogFormat offers you the ability to generate various logfiles, based on date, channel name, queue manager name, and name.

Options available:
LogFormat=
N - For name only or
M - For queue manager name only or
MN - Queue manager name  and name or
ND - Name and Date or
NDC - Name Date Channel format
NCD - Name Channel Date format or any variation

This option gives you the ability to generate a new file each day, which makes it easy to investigate who did what on a certain time.

**WARNING: Don't use this option on z/OS**

## Making BlockIP2 silent

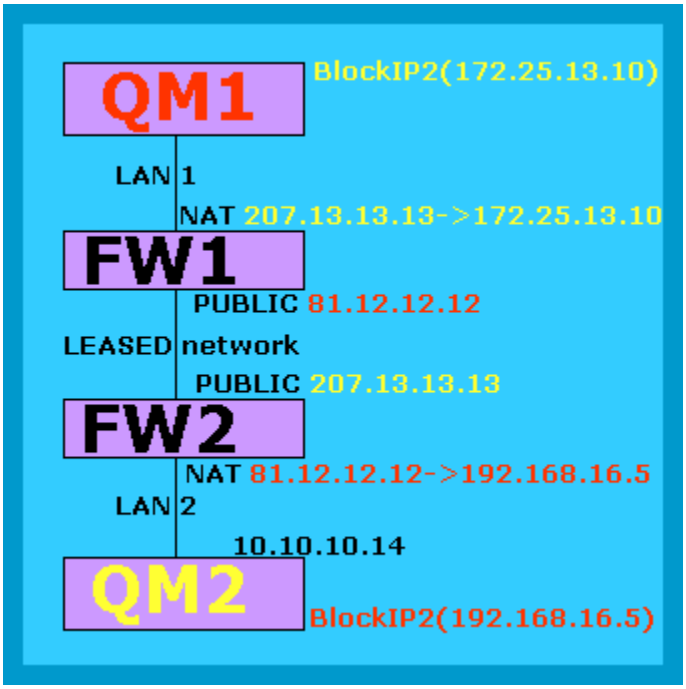
If you feel BlockIP2 floods your logs you can turn off the logging of connect and disconnect messages using the quiet (**-q;**) option in SCYDATA, see page 14 for more information.

## BlockIP2 and Firewalls

How do I configure BlockIP to handle communication using firewalls?

It's just as anything else, there are no problems in this area, just use the converted (NATed) address supplied by you network administrator.

I have a configuration like this, using two Queue Managers (QM1 and QM2) located in their own LAN, protected with Firewalls which can do NAT.




When I see QM1 from FW2 (marked **with red**), all I see is the public network address of FW1, in this case 81.12.12.12! I don't see the internal addresses inside LAN-1, they are hidden! Therefore I can only translate public address (If I need/like to), in this case I need to change it to a "local" address 192.168.16.5. This means that communication to and from QM1 in LAN-2 is done using 192.168.16.5, and that means that BlockIP2 should only allow 192.168.16.5 to get thru.

How do we typically start configure BlockIP2 for a new network connection?
This can be using either * (all allowed), or just block anybody, and study the log to find the wanted address. This is normally only necessary if your network specialist is unable to tell you how the incoming network is mapped....

Anyway your network specialists have to supply you with the address towards QM1....

## Filtering sequence

This section is added to help trouble shooting a filter.

BlockIP2 carry out the checks in this sequence:

In this order:

1. ProcessFile() See if configuration contains syntax errors.
2. CheckRunningChannels() See if channel limit is ok, if used.
3. CheckConnectionPattern() see if the presented IP-address is accepted.
4. CheckPartnerName() see if the presented partner name is accepted, if used.
5. CheckFAP() see if the presented FAP level is accepted, if used.
6. CheckBlockedUserId() see if we have a blocked user..
7. CheckUserId() see if user is in the positive list, if used.
8. CheckCONList() See if we have a CON= match, if used.
9. CheckSSLList() see if we have SSL= match, if used.
10. CheckBlankUser() check if we have a blank user..
11. CheckInvalidUsers() Check for special users...

See page 35 for more information about the keyword checking.

## Troubleshooting

In case of problems we will need the information:

- Platform: AIX, Solaris, windows incl. version.
- MQ version: output from dspmqver.
- Version of BlockIP2 incl. information about if it's a home compiled/modified version.
    - Include ls –l or DIR from the exit folder, to show the file size.
- The log file from BlockIP2, preferred from –d1; option.
- A brief description of the problem.
- In case of generated FDCs please include the ones related to the problem.

## Log file example

Example of the log:

```
2011-09-15|11:43:03|ConName is now [127.0.0.1]
2011-09-15|11:43:03|Starting on BlockIP2.ini processing
2011-09-15|11:43:03|Info: BlockIP2.ini [C:\IBM\WebSphereMQ\exits\BlockIP2.ini] to look for.
2011-09-15|11:43:03|QSJHPT02|QMGR= [QSJHPT02] matched [QSJHPT02].
2011-09-15|11:43:03|QSJHPT02|CHANNEL= [USESECPARMS] matched [USESECPARMS].
2011-09-15|11:43:03|QSJHPT02|BlockIP2.ini [C:\IBM\WebSphereMQ\exits\BlockIP2.ini] processing ended with rc 0.
2011-09-15|11:43:03|QSJHPT02|Finished BlockIP2.ini processing
2011-09-15|11:43:03|QSJHPT02|HandleCycleFiles() Entered
2011-09-15|11:43:03|QSJHPT02|  File size of C:\IBM\WebSphereMQ\exits\BlockIP2001.log is 174194 (limit 204800)
2011-09-15|11:43:03|QSJHPT02|HandleCycleFiles() Done
2011-09-15|11:43:03|QSJHPT02|======= INIT ======
2011-09-15|11:43:03|QSJHPT02|lMaxChannelActive is now -1 (before channel check)
2011-09-15|11:43:03|QSJHPT02|Return status Exitresponse=0
2011-09-15|11:43:04|QSJHPT02|HandleCycleFiles() Entered
2011-09-15|11:43:04|QSJHPT02|  File size of C:\IBM\WebSphereMQ\exits\BlockIP2001.log is 174607 (limit 204800)
2011-09-15|11:43:04|QSJHPT02|HandleCycleFiles() Done
2011-09-15|11:43:04|QSJHPT02|======= INIT_SEC ======
2011-09-15|11:43:04|QSJHPT02|ver=2.92 env=WIN ExitId=MQXT_CHANNEL_SEC_EXIT ExitReason=MQXR_INIT_SEC
ChannelType=MQCHT_SVRCONN
2011-09-15|11:43:04|QSJHPT02|BlockExit QMgr=[QSJHPT02] ChannelName=[USESECPARMS] ConnName=[127.0.0.1] Uid=[mrmq]
pDataLength=0
2011-09-15|11:43:04|QSJHPT02|BlockExit SCYDATA=[*;-d8;]
2011-09-15|11:43:04|QSJHPT02|CheckConnectionPattern() Entered
2011-09-15|11:43:04|QSJHPT02|  Pattern [*;*;]  ip[*] j 1
2011-09-15|11:43:04|QSJHPT02|  Pattern [*], ConName [127.0.0.1] to be checked next
2011-09-15|11:43:04|QSJHPT02|  Pattern [*], ConName [127.0.0.1] passed test..
2011-09-15|11:43:04|QSJHPT02|CheckConnectionPattern() Done
2011-09-15|11:43:04|QSJHPT02|Users: [] len [0] 0
2011-09-15|11:43:04|QSJHPT02|CheckCONList() Entered
2011-09-15|11:43:04|QSJHPT02|CheckCONList() Done
2011-09-15|11:43:04|QSJHPT02|CheckSSLList()
2011-09-15|11:43:04|QSJHPT02|CheckSSLList() Done
2011-09-15|11:43:04|QSJHPT02|CheckBlankUser() Entered
2011-09-15|11:43:04|QSJHPT02|CheckBlankUser() Done
2011-09-15|11:43:04|QSJHPT02|Connection accepted, Channel [USESECPARMS] ConName [127.0.0.1] Flags [IQNiL=1 ] User [mrmq]
2011-09-15|11:43:04|QSJHPT02|Return status Exitresponse=0
2011-09-15|11:43:04|QSJHPT02|HandleCycleFiles() Entered
2011-09-15|11:43:04|QSJHPT02|  File size of C:\IBM\WebSphereMQ\exits\BlockIP2001.log is 176082 (limit 204800)
2011-09-15|11:43:04|QSJHPT02|HandleCycleFiles() Done
2011-09-15|11:43:04|QSJHPT02|======= SEC_PARMS ======
2011-09-15|11:43:04|QSJHPT02|pWS->bUseSecurityParms 1 AuthenticationType 0
2011-09-15|11:43:04|QSJHPT02|Return status Exitresponse=0
2011-09-15|11:43:04|QSJHPT02|HandleCycleFiles() Entered
2011-09-15|11:43:04|QSJHPT02|  File size of C:\IBM\WebSphereMQ\exits\BlockIP2001.log is 176494 (limit 204800)
2011-09-15|11:43:04|QSJHPT02|HandleCycleFiles() Done
2011-09-15|11:43:04|QSJHPT02|======= SEC_PARMS ======
2011-09-15|11:43:04|QSJHPT02|pWS->bUseSecurityParms 1 AuthenticationType 0
2011-09-15|11:43:04|QSJHPT02|Return status Exitresponse=0
2011-09-15|11:48:58|QSJHPT02|======= TERM ======
2011-09-15|11:48:58|QSJHPT02|Channel closed [USESECPARMS] Connection Name [127.0.0.1]
2011-09-15|11:48:58|QSJHPT02|Before Free
2011-09-15|11:48:58|QSJHPT02|Free success
```

# Compilation

I've included the used compilation commands and option I use for building the exits. This should help you if you have another need for own compilation. For further information about how to compile please take a look in the IBM MQ Intercommunication manual. There is a chapter about compiling for various platforms here. This is for WebSphere MQ version 7.1 / 7.5 / 8.0.

### AIX and MQ version 7.1 / 7.5 / 8.0

```
xlc_r -q64 -e MQStart -bE:BlockIP2.exp -o BlockIP2
BlockIP2.c -I/usr/mqm/inc -D_REENTRANT -DUNIX -DHNLUP -
DAIX

cp ./BlockIP2 /var/mqm/exits64/.

chgrp mqm /var/mqm/exits64/BlockIP2

chmod 750 /var/mqm/exits64/BlockIP2
```

### Solaris_x86 and MQ version 7.1 / 7.5 / 8.0

```
gcc -c -m64 BlockIP2.c -ansi -fPIC -DUNIX -DSOLARIS -
D_REENTRANT -DHNLUP -I/opt/mqm/inc -Wall
/usr/ccs/bin/amd64/ld -64 -mt -G -o BlockIP2 BlockIP2.o -
R/usr/lib/64 -lsocket -lnsl -ldl

chown mqm:mqm BlockIP2

/bin/cp BlockIP2 /var/mqm/exits64

chown mqm /var/mqm/exits64/BlockIP2
```

### Solaris SPARC and MQ version 7.1 / 7.5 / 8.0 using gcc

```
gcc -I/opt/mqm/inc -m64 -o BlockIP2 BlockIP2.c -G -m64 -L
/usr/lib/64 -lthread -lsocket -lc -lnsl -ldl -DUNIX -
DSOLARIS -D_REENTRANT -DHNLUP
```

```
chown mqm:mqm BlockIP2

/bin/cp BlockIP2 /var/mqm/exits64

chown mqm /var/mqm/exits64/BlockIP2
```

### Solaris SPARC and MQ version 7.1 / 7.5 / 8.0 using cc

```
cc -xarch=v9 -xcode=abs64 -mt -G -o
/var/mqm/exits64/BlockIP2 BlockIP2.c -I/opt/mqm/inc -R
/usr/lib/64 -lsocket -lnsl -ldl -DUNIX -DSOLARIS -
D_REENTRANT -DHNLUP

chown mqm:mqm BlockIP2

/bin/cp BlockIP2 /var/mqm/exits64

chown mqm /var/mqm/exits64/BlockIP2
```

### Linux86_x64 and MQ version 7.1 / 7.5 / 8.0

```
gcc -m64 -o BlockIP2 BlockIP2.c -fPIC -ansi -shared -
D_REENTRANT -D UNIX -D_XOPEN_SOURCE -DHNLUP -Wall

chown mqm:mqm BlockIP2

/bin/cp BlockIP2 /var/mqm/exits64

chown mqm:mqm /var/mqm/exits64/BlockIP2
```

### Linux86 and MQ version 7.1 / 7.5 / 8.0

```
gcc -o BlockIP2 BlockIP2.c -fPIC -ansi -shared -D_REENTRANT
-D UNIX -D_XOPEN_SOURCE -DHNLUP -Wall

chown mqm:mqm BlockIP2

/bin/cp BlockIP2 /var/mqm/exits

chown mqm:mqm /var/mqm/exits/BlockIP2
```

### HP-UX and MQ version 7.0

```
c89 +DD64 +z -c -D_HPUX_SOURCE -o BlockIP2.o BlockIP2.c -
I/opt/mqm/inc

ld -b +noenvvar BlockIP2.o +ee MQStart +ee BlockExit -o
BlockIP2 -L/usr/lib/pa20_64 -lpthread

chown mqm:mqm BlockIP2

/bin/cp BlockIP2 /var/mqm/exits64

chown mqm:mqm /var/mqm/exits64/BlockIP2
```

### z/os and MQ version 7.0.1 / 7.1 / 8.0

See the included JCL in xxx.xmi

### z/Linux and MQ version 7.1 / 7.5 / 8.0

To be filled in.

### Windows and MQ version 7.0 / 7.5 / 8.0

```
cl -MT -c BlockIP2.c /DWIN32 /DWIN32_SHM /DHNLUP /FAscu -W3
-D__BI2_78__

link /DLL -out:BlockIP2.DLL BlockIP2.obj advapi32.lib
ws2_32.lib

cl BlockIP2S.c /MT /DWIN32 mqm.lib advapi32.lib
```

(It's expected that build path is extended to include WebSphere MQ libraries)

### OS/400

```
CRTSRVPGM SRVPGM(BLOCKIP2) EXPORT(*ALL)
```

# Messages

In the z/OS version of the exit have we added the following message id's to help automation, troubleshooting, and make documentation easier

| Message-id | Description |
|---|---|
| BLOCKIP-50I | Verification successful, connection accepted |
| BLOCKIP-51I | Verification successful, SSL have changes the userid, connection accepted |
| | |
| BLOCKIP-01E | Errors in SCYDATA specification. The specification doesn't match the required format. Check that SCYDATA() starts with 'FN=' or a valid pattern. |
| BLOCKIP-02E | Errors found in parameter file, check the extended error description |
| BLOCKIP-03E | Connection refused, User was found in the negative list. |
| BLOCKIP-04E | Connection refused, User was not found in positive list. |
| BLOCKIP-05E | Connection refused, ConName and User was not accepted in CON= specification |
| BLOCKIP-06E | Connection refused, The SSLPeer requested by CheckSSLList() was not successful |
| BLOCKIP-07E | Connection refused, for pattern [] user [] |
| BLOCKIP-08E | Connection refused, SSLPeerName too long max [] was [] |
| BLOCKIP-09E | Connection refused, CN requested blocked |
| BLOCKIP-10E | Connection refused for blank user identifier |
| BLOCKIP-11E | Connection refused for system user identifier (mqm) |
| BLOCKIP-12E | Connection refused for system user identifier (MUSR_MQMADM) |
| BLOCKIP-13E | Connection refused for system user identifier (musr_mqmadm) |
| BLOCKIP-14E | Connection refused, Pattern string is too long, max[] was [] |
| BLOCKIP-15E | Connection refused, Users string is too long, max[], was [] |
| BLOCKIP-16E | Connection refused, CON Spec string is too long, max [], was [] |
| BLOCKIP-17E | Connection refused, SSL Spec string is too long, max[], was [] |

| | |
|---|---|
| BLOCKIP-18E | Connection refused, BlockUsers string is too long, max[], was [] |
| BLOCKIP-19E | Error - Unknown Tag [] |
| BLOCKIP-20E | Connection refused, invalid file name pattern specified: FN=<filename...>; [] |
| BLOCKIP-21E | Connection refused, Specified file [] was not found |
| BLOCKIP-22E | Connection refused, Users string is too long, max[], was [] |
| BLOCKIP-23E | Connection refused, RespectMCA specified |
| BLOCKIP-24E | Connection refused for system user identifier (QMQM) |
| BLOCKIP-25E | Connection refused for system user identifier (CHIN task). |
| BLOCKIP-26E | Caching failed. |
| BLOCKIP-27E | Caching failed. |
| BLOCKIP-30E | Connection refused, Required ClientExit missing |
| BLOCKIP-31E | Connection refused, Wrong credentials supplied. |
| BLOCKIP-32E | Connection refused, Maximum number of channels was exceeded. |
| BLOCKIP-33E | Connection refused, number of SSL exceeds [] |
| BLOCKIP-35E | Connection refused, Channel [] ConName [] User [] PartnerName [] requested by CheckPartnerName() |
| BLOCKIP-36E | PartnerName check failed [] list [] |
| BLOCKIP-37E | Connection refused, PartnerName string is too long, max[], was [] |
| BLOCKIP-45E | Connection refused, Self signed Certificates are not allowed. Certificate was issued by [] |
| BLOCKIP-46E | Connection refused, Self signed Certificates filtering is not supported in pre version 6.0 |
| BLOCKIP-68E | Connection refused, CON Too many CON= specs active. Max [] |
| BLOCKIP-70E | Connection refused. Pattern NONE; was supplied, please check BlockIP2.ini |
| BLOCKIP-99E | wildcmp don't support mixed patterns yet. (0-J) |
| BLOCKIP-99E | No Matching SSL settings found for certificate: [] |
| BLOCKIP-99E | Connection refused, LOGCYCLE= invalid [%s]. Legal arguments are Y or N |
| BLOCKIP-99E | Connection refused, LOGCYCLE= semicolon(;) is missing |

| | |
|---|---|
| BLOCKIP-99E | Configuration queue is invalid [] name too long(). max |

## Considered changes

- Shared memory tool; to reset/manipulate the shared memory in case of problems.
- Implementing a chained configuration file feature for large customers.
- Reduce the file reading to gain performance on non z/OS systems, the catch is simplicity.
- Implement DNS caching to gain performance.
-

# Trademarks

# End User Agreement

You and your company are allowed to use and modify the program code as long that the code is not sold or build/included in a commercial product without a written accept from the author.

The intension of supplying the source code is to make it easier to WebSphere MQ professionals to keep their WebSphere MQ environments secure.

You are allowed to refer and/or put link on you web-site that points to http://www.mrmq.dk/BlockIP2.htm. You are also allowed to include this reference in your written documentation, but we are allowed to move the website and links without any notice.

You are **NOT** allowed to place the downloaded material on your own internet-webserver/homepage for download. Instead add a link to http://www.mrmq.dk/BlockIP2.htm.

**MrMQ.dk have no legal obligation if BlockIP2 causes your system any harm**, BlockIP2 is written and is delivered on AS-IS basis, and we're trying to keep it working. When you install it on your own system, it's like a tool written by yourself, and therefore on your own risk.

MrMq.dk is allowed to share received ideas and comments with our partners to enhance the WebSphere MQ products. Received traces and dumps are treated as confidential.

# Change History

| | | | |
|---|---|---|---|
| Written by<br>Jørgen Pedersen | | 12Dec02 | v1.00 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Improved due to inspiration from T.Rob, Sriniddhi and Peter Potkay<br>Now we allow more patterns specified, separated by semicolon (;), the number of patterns is limited by the size of SCYDATA field.<br>Protection against JMS intruders, or users using mqm or MUSR_MQADMIN. | 06Jan04 | v1.10 |
| Michael Dag | Changed / / comments to / * with ending ... as it wasn't suited for C compiler on AIX<br>Changed IFDEF to find out WIN32 or UNIX as UNIX can have 1 CPU as well<br>Commented the #pragma statement as this wasn't suited for C compiler on AIX<br>Removed from string and wild in the wildcmp routine and added [0] as it didn't work on Solaris | 20Jan04 | v1.11 |
| Jørgen H. Pedersen | Problem with simple pattern (172.22.22. ) this failed due to a design mistake, this is now solved. Now should the Exit work as designed.<br>Log improved using real date format, instead of internal formats...<br>Conditional compilation added, so the same source can be compiled on different platforms.<br>Some other improvements due to nature of UNIX/LINUX<br>so the log list is fine. | 22Jan04 | v1.12 |
| Jørgen H. Pedersen | Problem with missing timestamp in the log solved. Version 1.14.<br>Problem with pattern matching without trailing asterisk, this problem was introduced in 1.11. | 06Feb04 | v1.14 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Added functionality to verify the connecting userid, based on remote userid, this matching is also based on wild_cmp(), so wildcards is allowed. This pattern matching is case sensitive, so was51 is different from Was51. Check also added for right exit invocation. Any other exit will just be terminated.<br>inspiration:<br>SCYDATA('10.1.10. ;-n;userids=zz ,yy ')<br>SCYDATA('10.1.10. ;10.2.11. ;userids=michael,zz ,yy ') | 10Feb04 | v1.15 |
| Jørgen H. Pedersen | Added functionality to allow specification rules to be passed in a file, this file is specified in SCYDATA() like (C:\path\rulefile.txt) or (/var/mqm/exits/rulefile.txt) depending on your platform.<br>This allows you to have a rule file per channel, and this offers flexibility to implement various security schemas.<br>Two extra parameters have been added to SCYDATA: -d & -q. -d is used for debugging purposes, while -q is q Quite mode, where the output is very reduced (only one line per attempt to connect). | 15Feb04 | v1.16 |
| Jørgen H. Pedersen | A serious user-validation error was introduced in 1.16 which only allowed usage of FN= mode.<br>The error is corrected (i hope), and my tests shows it works.<br>Enhanced error reporting and control of filters.<br>And still be restrictive so an nRetCode don't open the connection. | 16Feb04 | v1.17 |
| Jørgen H. Pedersen | The exit is now able to allow blank lines and empty lines. | 20Feb04 | v1.18 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | The exit is extended to display SSL properties.<br>Pattern matching for x503 development started.<br>Specifications are passed in configuration file.<br>No test are done if options != "FN=" due to performance reasons.<br>This filter option is combined with an DN->User generic mapping function to ease systems administration, so unknown users with valid certificates can be kept outside our world.<br>The way the SSL-specs are searched is top down and with stop on first match, this gives the ability to "shop" in the CN's, and just take the ones we like and block the rest.<br>Following options are implemented:<br>SSL=CN=ibmwebspheremqQSJHPT01,MCA=user;<br>change MCAUSER for this CN<br>SSL=CN=ibmwebspheremq ,MCA= ;<br>allow all connections from ibmwebsphere<br>SSL=CN= ;BLOCK;<br>Refuse all connections<br>SSL=CN= ;MCA=NoBody; | 20Feb04 | v1.20 |
| Jørgen H. Pedersen | The logic of the exit have been changed so it's no longer possible to penetrate this exit using a exit on the client end, to send a sec_msg, to get a connection accepted status.<br>This design mistake would leave the channel open to almost anybody, that was capable of creating a client security exit. | 05Mar04 | v1.21 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Added functions to specify connectionname/userid match, so we can combine the features of security. This matching feature only works on FN= mode, where the connectionname/userid is specified using the CON= keyword:<br>CON=<conname>;<userids>[;MCA={ \|userid}];<br>CON=10.11.12. ;u ;MCA=sysoper;<br>CON=192.162.1.14;zz ;<br>The list is searched for first match of connectionname+userid, which means it's very important to specify the options in the right order.<br>If you specify:<br>CON= ; ;MCA=root;<br>CON=162.12. ;peter;<br>This will have the result that all incoming connection attempts will have the MCAUSER forced to "mqm", even "peter", because all connetions+userids will match ; ;. You see the point? | 20Mar04 | v1.22 |
| Sid Young | Complete re-write and restructure of code.<br>Changed logic to deny all and only accept if all conditions required are met<br>Added enhanced logging capability<br>Changed BlockExit() to call separate init, parse and test methods. | 22Mar04 | |
| Sid Young | Added code to support channel and user name stamping in log file name.<br>Found type in ProcessLine(). Tested OK. | 29Mar04 | |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Retrofitted some code to create a windows version of the program that could be loaded. And to allow capability with old versions that works without a rule-file. Restrictions added so connections will be refused if failure in rule/parameter specifications, because my auto test passed many connection attempts that should be blocked due to rules.<br><br>This is a valid rule spec:127 ;userids=u ;-q; resulted in : WarningUnknown Tag [127 ;userids=u ;-q;] and gives connection accepted .... :o(<br>Problem with blank rule lines -> gives connection blocked.<br><br>Exit changed so it respects commented lines containing valid commands like: #BlockMqmUsers=Y; which in the redesigned version was taken as a command.<br><br>This was caused by missing init of variables if the exit was redriven, so the state was saved between invocations on the same connection.<br><br>Specification of logfiles changed to conform with the used standard where all specs. are terminated with a semicolon.<br><br>Small changes where numeric constants have been changed to WMQ supplied constants. | 30Mar04 | v2.10 |
| Jørgen H. Pedersen | Some editorial changes to allow compilation under AIX without problems.<br>UNIX logging changed, so incorrect log specification information still will result in logging, where the spec. is bypassed. LogDirecty removed and is replaced with LogPath, for compliance between environments. | 27May04 | v2.11 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Reporting of refused connections added to log, so we can see who is trying to get in, this was disappeared on Patterns= and CON= mismatch this is fixed so we can continue. UseridUpperLowerCase= Introduced. Default is case sensitive to be compatible with older versions. To ignore case use UseridUpperLowerCase= . Any other is invalid. BlockUsers=, added, so we now can use a negative list, but still with precedence of CON=, and SSL=, meaning that we can change an incoming user to a blocked user! | 29May04 | v2.12 |
| Jørgen H. Pedersen | Validation of SCYDATA field is enhanced, so more errors are reported, for investigation. | 07Jun04 | v2.13 |
| Neil Casey | No change to functionality at all, but added Windows compiler information so that external link directives are not required, and added MVS directives to allow generation of an MVS targeted exit. Files specifications on MVS can be //DD:JCLNAME or UFS hierarchical names (i.e. unix names) Also fixed issues with ConnectionNames. The code was using the EXIT_NAME_LENGTH to get a value for the length of the ConName. This doesn't work on MVS where the exit name length is 8. Change all references to use the MQ defined constant MQ_CONN_NAME_LENGTH. | 03Aug04 | v2.13 |
| Jørgen H. Pedersen | Validation of SCYDATA field is enhanced, so errors on specifying FN=file is enhanced. If KW FN= not specified, it's mandatory that SCYDATA contains either a asterisk ( ), question mark(?) or a pattern starting with 0-9. Due to the fact that the latest problems reported is in this area of bad specification.<br>Hard-brackets ([]) is changed in z/OS implementation to (<>), so it becomes readable in non-English environments. | 05Aug04 | v2.14 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Neil Casey | Extended maximum pattern lengths to 256 characters. Changed logic for building the pattern strings so the multiple Pattern= or User= etc lines add to the previously built string, instead of overwriting it. This is a function change which makes this version behave differently when faced with multiple lines of data which previously just used the last data found. | -10Aug04 | v2.13 |
| Jørgen H. Pedersen | just a small one... together with a small code change for Solaris so that the timestamp is printed too. This is done using wcsftime... | 11Aug04 | v2.15 |
| Neil Casey | Remove the wcsftime call. The parameter passed to it (char ) by the code did not match the expected parameter (wchar ) The time stamping works fine on Solaris without this mod. Remove some commented out code which was for debugging.<br>Reformat source indenting. | 24Aug04 | v2.16 |
| Jørgen H. Pedersen | A small change about setting MCAUSER based on CON= control, z/OS will have a field filled with trailing spaces. Added RespectMCA keyword. | 08Feb05 | v2.17 |
| Jørgen H. Pedersen | Handling of SEC_PARMS added as part of WMQ 6.0 support. | 21Apr05 | v2.18 |
| Jørgen H. Pedersen | Added support for max connections on a given channel.<br>New keyword:<br>MAXCHL=ChannelName;MaxConnections; was added.<br>This is currently not supported on WebSphere MQ for Z/OS. | 24Apr05 | v2.20 |
| Jørgen H. Pedersen | Problem with userid check in CON= Fixed.<br>Wildcard compare extended with character range, numeric, alpha patterns, to enhance filtering capabilities.<br>AllowBlankUserID added, Default is changed so we don't allow blank userids anymore. | 03May05 | v2.21 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | MAXCHL=ChannelName; MaxConnections; is now supported on WebSphere MQ for Z/OS prior to version 6.0. Using MQSC: "DISPLAY CHS(<channel name>) CURRENT" command.<br>Explanation on exit-reason and channel type enhanced, to give the user a better understanding of what is happening. We have fixed the known problem in the userid-list of the CON= specification so it now works as designed.<br>On z/OS we're also blocking connection attempts using the STCID from the xxxxCHIN task, this should prevent unauthorized access. | 17May05 | v2.22 |
| Jørgen H. Pedersen | Problem with reentrant code changed, so it should be reentrant.<br>Syntax check of parameter specification is improved. | 1Jul05 | v2.30 |
| Jørgen H. Pedersen | strtok changed to strtok_r for non windows implementations to deal with reentrant code problems. This should help make it more stable. | 11Sep05 | v2.31 |
| Jørgen H. Pedersen | MAXCHL for unix changed to use shrmem for performance reasons. | 20Sep05 | v2.32 |
| Jørgen H. Pedersen | OS/400 support added (bypass strtok_r) and handling a new print model.<br>New OS/400 entrypoint (int main() added with conditional compile. | 25Oct05 | v2.35 |
| Jørgen H. Pedersen | Handling of semaphores in LockSpecSem() fixed. | 14Nov05 | v2.36 |
| Jørgen H. Pedersen | Connection refused message enhanced to conname and channelname | 14Nov05 | v2.37 |
| Jørgen H. Pedersen | Connection refused, Pattern string is too long, max. This message wasn't showing the resulting message length, just the current one. So when concatenate patterns we had a issue. This was also fixed for Userids | 12Dec05 | v2.38 |
| Jørgen H. Pedersen | Porting for Linux AMD64 various definitions changed. | 25Jan06 | v2.39 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Problem with semaphore addressing fixed. And a bit about logging on UNIX.<br>shmdt(shm); changed to shmdt((char )shm);<br>due to changes on Solaris x86 | 5Feb06 | v2.40 |
| Jørgen H. Pedersen | Problem when a configuration file contains garbage and MQXR_TERM is<br>invoked and it returns a failure .... This was seen WIN2K3. | 13Feb06 | v2.41 |
| Jørgen H. Pedersen | Logging to EventLog added on windows.<br>Currently all events are logged.<br>Switch will be added to SCYDATA to control this feature. | 14Feb06 | v2.42 |
| Jørgen H. Pedersen | Needed cleanup procedure for getActualCurrentNumberOfRunningChannels to cleanup allocated "databags" mqDeleteBag must be included in the windows edition to prevent storage leak. | 18Feb06 | v2.43 |
| Jørgen H. Pedersen | Added support for shared memory on windows.<br>This is implemented using a separate program BlockIP2S, that initializes the Shared Memory Segment(SMS). There are a detached BlockIP2S per queue manager running with BlockIP2. (When the channel limiter is activated).<br>Storage Leak from pre. Version 2.44 is also solved. This applies only to the delivered windows DLL, due to a compilation problem.<br>Serialization was added in windows to handle logfile contention. This is also included in the control of SMS and BlockIP2S. | 1Mar06 | v2.44 |
| Jørgen H. Pedersen | Shared memory naming changed to support HACMP and other MA special settings.<br>The shm name under NIX is now based on /var/mqm/mqs.init | 19Mar06 | v2.45 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Added logic to handle cycling of logfiles, to prevent fill up of filesystems. Controlled by: LogCount=nn; # of versions (between 3 and 99). LogSize=nnnnnnn; Size and the logfile before switching. min 100KB. Default LogDrive log LogPath log path is changed on windows to conform with the UNIX implementation, so we use the windows settings. And FileName is extended with "001" to allow circular logging. | 4Apr06 | v2.46 |
| Jørgen H. Pedersen | Problem with Log file rotation solved. Error messages were enhanced. | 5Apr06 | v2.48 |
| Jørgen H. Pedersen | Pattern matching extended to allow imbedded this means that the generic specs may look like this: Patterns=123. .123; SSL=CN=ibmwebspheremq T01,MCA=user; ASC=Y/AllowSelfSignedCertificates=Y added. SSL=[C=,][L=,][O=,][OU=,]CN=;[[MCA={userid\|role\| };]\|[BLOCK;]] some examples: TERM=N/Y for controlling print of termination message. MAX_SSL raised to 256 and MAX_PL to 1024 4. | 20Apr06 | v2.50 |
| Jørgen H. Pedersen | wildcmplist() problem fixed | 20Apr06 | v2.51 |
| Jørgen H. Pedersen | messlen and buflen changed from long to MQLONG in deductStatusQ, getActualCurrentNumberOfRunningChannelszOS | 20Apr06 | v2.52 |
| Jørgen H. Pedersen | Support for BLANK_USERID added, changes was made in CheckUserId() and CheckCONList() to obtain the wanted functionality. Testcase: test25a-b, test39a-b. Implementation of hostname support done in CheckConnectionPattern() This implementation does not support IPv6. Deleted many compile warnings to get a clean compilation list without warnings. | 10May06 | v2.53 |
| Jørgen H. Pedersen | Storage leak on z/os solved together with DNS lookup on z/os. | 28May06 | v2.55 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Problem with errors in spec. files solved.<br>AMQ9190 The user exit ... invoked for .. with id '11' and reason '12', returned values that are not valid, as reported in the preceding messages. The channel stops.<br>Detection for IY86343. added. | 10Aug06 | v2.56 |
| Jørgen H. Pedersen | Problem with accepted CON= where there are no CON= that should give the auth. | 10Aug06 | v2.57 |
| Hubert Kleinmanns | Fixed a problem in function 'CheckSSLList'. BlockIP2 exited in this function with a zero length 'SSLRemCertIssNamePtr' in structure 'pChannelExitParams'. This problem occurred on Solaris Sparc systems with MQv6. | 10Oct06 | v2.56a |
| Jørgen H. Pedersen | Implementations of additional specification file for configuration.<br>BlockIP2 look default for /var/mqm/exits/BlockIP2.ini or ExitPath\BlockIP2.ini on the distributed platforms if FN= is not specified. This is implemented to ease distribution in complex installations so a generic specification can be used. The NEW syntax on the BlockIP2.ini is documented in the manual.<br>If the file is not present or cannot be read due to access violation it's treated as not found and BlockiP2 will continue as before. | 10Dec06 | v2.60 |
| Jørgen H. Pedersen | Differentiated debug modes added (-d0, -d1 and -d8) and -i for ignore looking for BlockIP2.ini<br>-d0 is typically used for debug of the spec file.<br>-d old mode, lists all exitpoint invocations.<br>-d1 shows functions called (technical debug).<br>-d8 Code development debugging mode)<br>LOGCYCLE={N\|_Y}; added to get rid of 001 logfiles to back to behavior pre 2.46<br>-l option added to help LOGCYCLE, from SCYDATA. | 12Jan07 | v2.61 |
| Jørgen H. Pedersen | NIX problem in ProcessMQSiniFile_qm_ini solved when looking for BlockIP2.ini<br>Problem with file cycle solved for dist. platforms. | 16mar07 | v2.62 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Complex SSL filtering problems solved. Added ST= and PC= in the SSL filtering to be compliant with gsk7cmd and runmqckm and their capabilities.<br>Problem with FNx= fixed. Reason was premature release or storage. | 05Apr07 | v2.64 |
| Jørgen H. Pedersen | Added MCA=xx(nn,nn) capabilities on SSL= to enhance the extraction of an userid that can be used for authentication. Any DN keyword can be used as key....<br>The function works "substring like" starting with index 1 as first position and with "end of string" when next separator "," are reached.<br>SCYDATA() written to log when using logmode -d0 and up.<br>NIX storage leak fixed. localtime() changed to localtime_r() to prevent highly used system to breakdown. The use of gethostbyname() changed to gethostbyname_r() to prevent SIGSEGV. | 06May07 | v2.65 |
| Jørgen H. Pedersen | Problem with channellimitter solved when using BlockIP2.ini | 17May07 | v2.66 |
| Jørgen H. Pedersen | Problem with FN= and file not found abend solved.<br>Problem with QMGR= and CHANNEL= logic solved. CHANNEL= stmt didn't lock for previous accepted channel.<br>ENV changed to show platform information. Like MVS, AIX, Linux, WIN etc.<br>Added #pragma to disable _POSIX warning 4996 for fileno. | 06Jun07 | v2.67 |
| Jørgen H. Pedersen | Support for AS/400 implemented... And it's working....<br>Extended to use shared storage and DNS Support. | 24Jun07 | v2.68 |
| Jørgen H. Pedersen | Sporadic errors solved for Solaris and others<br>Limit on CON_MAX enforced BLOCKIP-68E added. | 11Jul07 | v2.69 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Security problem with windows 2003 solved, changed in BlockIP2 and BlockIP2S.<br>Extended error reporting in starting windows SHM and registry load.<br>SYSLOGFCLTY= and SYSLOGPRTY= added for UNIX to control syslog().<br>syslog_r() added for AIX for better thread safe.<br>Show the name of the BlockIP2.ini in -d option, to ease configuration trouble.<br><br>Report open reason for failed file operations..... (fopen() errno)<br>BlockIP2 failed to open the specified logfile [] and other. | 12Mar08 | v2.70 |
| Jørgen H. Pedersen | Changed some sprintf() to snprintf() to avoid stack crashes.<br>Added thread id to identify the correspondent requests and answers. | 20Oct08 | v2.71 |
| Jørgen H. Pedersen | Added CloseHandle(mutex) to prevent loss of handles, and to keep the stuff running for long periods of time. | 21Oct08 | v2.72 |
| Jørgen H. Pedersen | Changed size of connection table size to 64KB from 2, giving room for 1500 channels. | 25Oct08 | v2.73 |
| Jørgen H. Pedersen | Changed LOGDEBUG1 to LOGDATA for nospace in chltable for UNIX implementation.<br>Channel structure changed to BlockIP2.2a to assure compliance with old version, and contains now SHMMAX_CHL. | 30Jan09 | v2.74 |
| Jørgen H. Pedersen | Suppress TERM note when quiet mode selected.<br>AllowBlankUserID=N added to comply with the book.... | 12Mar09 | v2.75 |
| Jørgen H. Pedersen | Added support for longer WTO messages on z/OS. Problem with BLOCKIP2-I50 and CONNAME print solved. z/OS will now allow change of channel limiter threshold after first time usage.<br>Added support for WTOPFX under z/OS for WTO's to comply with CA tools... | 20Mar09 | v2.76 |
| Jørgen H. Pedersen | Support for allowing MCAUSER specified on the channel to be overridden by the incoming userid, this is done with the MCA=* on the CON= statement. | 22Jun09 | v2.77 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | BlockMqmUsers=N added to comply with the book....<br>Problem with CON= and hostnames solved. It username and MCA was removed due to logic error.<br>Need for mqm linkage removed for most platforms. | 07Jul09 | v2.78 |
| Matt Batterham | Support for an empty SCYDATA field.<br>Empty SCYDATA() defaults to “NONE;” | 06Oct09 | v2.79 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Problem with MQ 7.0.1.0 (AIX) where conname can contain the source port: 10.1.1.1(1490)<br><br>LogFormat=M (Queue Manager) added to include the queue manager in the logfile name.<br><br>QMGR_SHORT_NAME=<short name>; up to 8 characters short name of qmgr to include in the logfile, useful when long queue manager are used... By default it takes the first 8 characters from queue manager name.<br><br>INCQMGR=Y; added to include a short qmgr id in the logfile. Highly useful for large systems.<br>Not so useful when used with LogFormat=M.<br><br>Introduced USESECPARMS=Y; that allows the user to set the MCAUSER based on the content in SecurityParms. This allows users to use BlockIP2 with MQExplorer to do authentication using an OAM exit.<br>**NOTE** If not used together with an Object Authority Manager (OAM) this leads to security exposure because to example: MQExplorer (and other applications) allow the user to type in an userid aka mqm.<br>Default is USESECPARMS=N; so it's your own choice to enable the feature.<br>However it requires the invoking client application to utilize MQCSP_AUTH_USER_ID_AND_PWD.<br>From the manual: This value indicates that MQCSP user ID and password fields will be used by the Object Authority Manager (OAM) to perform authentication on a MQCONNX call. When this is specified, the MQCSP structure is passed to the OAM Authenticate User function, which can set appropriate identity context fields. Designed by Duke Nguyen.<br><br>AllowSelfSignedCertificate= has been removed from the manual. But are kept in BlockIP2 logic for capability.<br>AllowSelfSignedCertificates= is the right one. | 08Dec09 | v2.80 |
| Jørgen H. Pedersen | Problem with MQ 7.0.1.0 where conname can contain the source port: 10.1.1.1(1490) | 14Dec09 | v2.81 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Problem with channel limiter on heavily used multithreaded UNIX where shared memory was exhausted. Changes made to wildcmplist() to solve problem with "BLANK_USERID" in a multiline specification: "CON=*;BLANK_USERID,QMQM,QMQMADM,QSECOFR;MCA=NoBody;" A dedicated test is included to deal with this test. LogCycle= added to ease configuration.<br><br>Added debug level on ver=x.xx line for ease support. | 03Feb10 | v2.82 |
| Jørgen H. Pedersen | Added support for multi reply from DNSservers Added support for standby qmgrs. (DataPath on UNIX) | 04Mar10 | v2.83 |
| Paul Giordano | Add logic to z/OS to load process parameters into memory for efficiency. Keyed with nZOSLoadMem and parm -m. Added Silent flag to 50I and 99I WTOs. Removed unused __4kmalc() and #if BLOCKIM logic. Retrofitted by MrMQ. | 06May10 | v2.84 |
| Jørgen H. Pedersen | Problem with SIGSEGV when processing mqs.ini solved. Was caused by storage overlay. Problem with missing log files solved (was also causing SIGSEGV) thanks to Anshul Rastogi and others. | 16Aug10 | v2.85 |
| Jørgen H. Pedersen | More changes regarding SIGSEGV done. | 18Aug10 | v2.86 |
| Jørgen H. Pedersen | More changes to support standby queue managers. Monitoring failover. UNIX Shared memory anchor point moved from mq path to /tmp/.<qmgr>_BlockIP2_<fid> to prevent further problems with MQ product changes. This can be disabled with option -t, or TMPPATH=No; in configfile. TMPPATH=Yes; Is default. | 22Jan11 | v2.90 |
| Falk Dressler | Avoid to refer to invalid memory when checking for '\n' in configuration file lines | 14Mar11 | v2.91 |
| Jørgen H. Pedersen | Problem with CLUSSDR channels and "empty" patterns solved. | 12Apr11 | v2.92 |

| ***Changes History*** | | ***Date*** | ***Version*** |
|---|---|---|---|
| Jørgen H. Pedersen | Enhancement of debugging for option USESECPARMS=Y | 13Sep11 | v2.93 |
| Jørgen H. Pedersen | Enhancement of debugging for blank userids and exit response.<br><br>Additional logic added to control the behavior of SEC_PARMS, to allow DotNet applications to connect to MQ without formal authentication and to respect the MCAUSER settings. | 20Sep11 | v2.94 |
| Jørgen H. Pedersen | Changed determination of ENV to have right identification of environment. Added ZLINUX conditional definition | 26May12 | v2.95 |
| Jørgen H. Pedersen | Removed need for including CSQXSTUB in linkage step to prevent IEW2456E complaining about missing MQOPEN/MQCLOSE/MQGET and MQPUT1 modules to remove the need for using a PDSE as CSQXLOAD. Perhaps all this old stuff should be removed to clean up the source. | 15Jun12 | v2.96 |
| STB | Replaced LoadRegistry() to automatically support multi install under windows. Retrofitted by MrMQ. | 10Sep12 | v2.97 |
| Jørgen H. Pedersen | Added FAP= check parameter to block connections from older MQ devices, like MQ 5.x clients. | 03Jan13 | v2.98 |
| Jørgen H. Pedersen | Forced initiation of nZOSLoadMem assure configuration is read on z/OS V2R1 | 30Oct13 | V2.99 |
| Jørgen H. Pedersen | Added support for PartnerName checking to assure identity of connecting partner<br>Added support for IBM MQ version 8.0 on Windows (64 bit)<br>Updated z/OS connection limiter. | 09Mar15 | v3.00 |