https://www.ibm.com/support/pages/node/595111

CHLAUTH Made Simple:
Common Scenarios and Examples and
How to Verify them with RUNCHECK

Date last updated: July 27,2020

Mike Cregger – mike_cregger@us.ibm.com
IBM WebSphere MQ Support

+++ Objective

The objective of this technical document is to provide examples how to use Channel Authentication (CHLAUTH) rules to better control access to your WebSphere MQ queue managers. Common problems caused by CHLAUTH rules are noted, along with examples of CHLAUTH rules to control access.

Added Scenario 7 – CHLAUTH for RCVR channels, SDR/RCVR pairs.
(See Table of Contents)

Table of Contents

Overview of CHLAUTH:

With the advent of Channel Authentication (CHLAUTH) rules in WebSphere MQ (WMQ) v7.1 and higher, there is new behavior/functionality when connecting to a queue manager. The default behavior for new queue managers is changed, such that CHLAUTH may not allow users to connect, that could connect pre v7.1.

CHLAUTH rules are used to determine if a channel can be started, and they allow mapping via MCAUSER to another userid. Thus, it is possible (even likely) that the channel cannot start due to the result of the CHLAUTH rules; this has caused many many problems. There are several technotes and some developerworks articles on this subject in the additional references section at the end of this document.

In this techdoc, we will discuss using the CHLAUTH features and common scenarios you may want to employ in your WMQ environment. We will provide examples of how to use CHLAUTH to your benefit, based on questions and service requests we have had from you.

We added CHLAUTH rules because many people wanted better control over what users were accessing WMQ, the ability to control this by channel, without the need to use a custom security exit.

The default behavior for new queue managers is changed with WMQ v7.1 and higher as CHLAUTH rules are enabled by default. If a queue manager is migrated from a version prior to v7.1 (ie: not created fresh with crtmqm), CHLAUTH will be DISABLED in the queue manager properties. Many folks migrate to new versions and new hardware and use scripts to rebuild their queue managers. In this case, if CRTMQM is used, CHLAUTH will be enabled and in effect by default.

As part of CHLAUTH, there are 3 new default CHLAUTH rules for new queue managers. The change in behavior caused when these 3 default rules are enabled is the main cause of problems.

Result of 3 default CHLAUTH rules:

- NO ACCESS to all Channels by any MQ-admin* users
- NO ACCESS to all SYSTEM.* channels by all users
- ALLOW access to SYSTEM.ADMIN.SVRCONN (default channel used by WMQ Explorer)

*MQ-admin = special group which allows MQ adminstrative privileges, (ie: mqm group in Unix)

CHLAUTH rule processing determines which rule to use, based on the following precedence. More specific rules take precedence over generic or less specific rules.

For example, the 3 default rules, 2 of them:
" NO ACCESS to all Channels by any MQ-admin* users" and
" NO ACCESS to all SYSTEM.* channels by all users"
lock down access to channels, but the 3rd rule
"ALLOW access to SYSTEM.ADMIN.SVRCONN"
is more specific, and thus takes precedence over the other 2 if the channel is the SYSTEM.ADMIN.SVRCONN channel; thus allowing access on that channel.

How to display CHLAUTH rules:

You can display the CHLAUTH rules, by entering runmqsc and using the DISPLAY CHLAUTH command.
(example from new queue manager, 3 default rules)

runmqsc: DISPLAY CHLAUTH (*) ALL

```
    1 : display chlauth (*) all
AMQ8878: Display channel authentication record details.
  CHLAUTH(SYSTEM.ADMIN.SVRCONN)         TYPE(ADDRESSMAP)
  DESCR(Default rule to allow MQ Explorer access)
  CUSTOM( )                             ADDRESS(*)
  USERSRC(CHANNEL)                      ALTDATE(2013-05-23)
  ALTTIME(13.54.19)
AMQ8878: Display channel authentication record details.
  CHLAUTH(SYSTEM.*)                     TYPE(ADDRESSMAP)
  DESCR(Default rule to disable all SYSTEM channels)
  CUSTOM( )                             ADDRESS(*)
  USERSRC(NOACCESS)                     WARN(NO)
  ALTDATE(2013-05-23)                   ALTTIME(13.54.19)
AMQ8878: Display channel authentication record details.
  CHLAUTH(*)                            TYPE(BLOCKUSER)
  DESCR(Default rule to disallow privileged users)
  CUSTOM( )                             USERLIST(*MQADMIN)
  WARN(NO)                              ALTDATE(2013-06-03)
  ALTTIME(09.37.10)
```

With this new functionality and default CHLAUTH rules in place, we commonly see errors that may be caused by CHLAUTH rules, especially as customers utilize the newer versions of WMQ more and more.

Common connection errors which can be due to CHLAUTH rules:

CHLAUTH rules are used to determine if a channel can be started, and they allow mapping via MCAUSER to another userid. If the channel can not be started due to the CHLAUTH rules, the following errors are commonly seen.

RC 2035 MQRC_NOT_AUTHORIZED
RC 2059 MQRC_Q_MGR_NOT_AVAILABLE
AMQ4036 Access not permitted
AMQ9776: Channel was blocked by userid
AMQ9777: Channel was blocked
MQJE001: An MQException occurred: Completion Code 2, Reason 2035
MQJE036: Queue manager rejected connection attempt

The resulting MCAUSER is then used for checking MQ object authority.
There are several ways to diagnose and resolve these and several possible work-arounds. You can work-around the issue, or take control and create CHLAUTH rules which suite your needs.

Best Practices for CHLAUTH:

Best practices would state to use CHLAUTH rules to improve your overall security and management of MQ access.

It is considered a best practice to be more secure and block access strictly; then add more specific CHLAUTH rules, to control who can access/start the channels (see example in scenario 5 below.) This is discussed as a “back-stop” rule (link at the end in the additional resources section.)

See scenarios below to modify/add rules to address access issues. We will present work-arounds, but it is better to utilize the default rules to control access.

Work-around 1 - Disable CHLAUTH:

As a work-around and also to troubleshoot the errors above, you disable CHLAUTH rules. They can be re-enabled any time. If disabling the CHLAUTH resolves the connection issue, you know that this was the cause. Overall use of CHLAUTH is enabled/disabled at the queue manager level, with the CHLAUTH property.
As CHLAUTH rules are new, and didn't exist prior to v7.1, many folks disable them, if they feel the added security is not needed, especially if wanting the same functionality and access prior to CHLAUTH rules.

To disable CHLAUTH rules, alter the CHLAUTH property on the queue manager.

runmqsc: ALTER QMGR CHLAUTH (DISABLED)

You can also set it to WARN, which would allow access, but still log the result of the rule.

Work-around 2 - Modify or Remove CHLAUTH rules:

You can also delete or modify the CHLAUTH rule(s) causing your problem.

To modify a CHLAUTH rule, you use the SET CHLAUTH command with the ACTION (REPLACE).

Example runmqsc command to MODIFY the default rule which causes "NO ACCESS to all Channels by any MQ-admin users" to WARN, instead of blocking:

runmqsc: SET CHLAUTH (*) TYPE (BLOCKUSER) USERLIST (*MQADMIN) WARN(YES) ACTION (REPLACE)

To delete a CHLAUTH rule, you use the SET CHLAUTH command with the ACTION (REMOVE).

Example runmqsc command to DELETE the default rule which causes "NO ACCESS to all Channels by any MQ-admin users":

runmqsc:
SET CHLAUTH (*) TYPE (BLOCKUSER) USERLIST (*MQADMIN) ACTION (REMOVE)

Testing access using MATCH (RUNCHECK):

You can test the result of your CHLAUTH rules, using the MATCH (RUNCHECK) option of the CHLAUTH rule in runmqsc. MATCH (RUNCHECK) option returns the record that will be matched by a specific inbound channel at run time if it connects into this queue manager. You must provide:
- the channel name
-ADDRESS attribute
- SSLPEER attribute, only if the inbound channel will use SSL or TLS
- QMNAME or CLNTUSER attribute, depending on whether the inbound channel will be a client or queue manager channel, you must provide a CLNTUSER

Example of checking what CHLAUTH rule will result in user 'mcregge' accessing a channel named "CHAN1". 'mcregge' is a MQ-admin user, with the default rules in place:

runmqsc: DISPLAY CHLAUTH (CHAN1) MATCH (RUNCHECK) CLNTUSER ('mcregge') ADDRESS ('192.168.1.138')

AMQ8878: Display channel authentication record details.
CHLAUTH(*) TYPE(BLOCKUSER)
USERLIST(*MQADMIN)

For 'mcregge', the Channel will NOT run, the user will be blocked due to the BLOCKUSER rule for *MQADMIN users.

Example of checking what CHLAUTH rule will result in 'alice' accessing a channel named "CHAN1". 'alice' is NOT a MQ-admin user, with the default rules in place:

runmqsc: DISPLAY CHLAUTH (CHAN1) MATCH (RUNCHECK) CLNTUSER ('alice') ADDRESS ('192.168.1.138')

AMQ9783: Channel will run using MCAUSER('alice').

For 'alice', the Channel will run and it will pass 'alice' in as the MCAUSER. The MCAUSER is the userid used to check MQ object authorities.

New option in MQ v9.2+, Ignore case when matching incoming client user id:

A new option was added which can be enabled via the qm.ini

ChlauthIgnoreUserCase = YES

Enables a queue manager to make incoming client userid matching within CHLAUTH rules case-insensitive. This option allows the:

- CLNTUSER in CHLAUTH TYPE(USERMAP) rules to be matched case-insensitively
- USERLIST in CHLAUTH TYPE(BLOCKUSER) rules to be matched case-insensitively

Resolve the issue by creating new CHLAUTH rules:

As noted the best practice would be to use CHLAUTH rules to control who is accessing your WMQ queue managers. This is done by ensuring you have appropriate CHLAUTH rules in place.
In this section, I'll provide some common scenarios requested by customers, and example CHLAUTH rules to accomplish these. I prefer to start with the 3 default rules, as provided by the WMQ product for a new queue manager.

Scenario 1: Control access for specific MQ-admin users

A server connection channel that is to be exclusively used for an administrative perspective. ie to connect from MQ Explorer. We have a specific channel for this usage and defined IP address(es) from where we want connections to only accept from but should also block access to 'mqm' ID if connection is not from the specified IP address.

Make a SVRCONN channel to be used for this scenario.
ADMIN.CHAN - For MQ Explorer, MQ-admin users

runmqsc: DEFINE CHANNEL (ADMIN.CHAN) CHLTYPE (SVRCONN) TRPTYPE (TCP)

For testing, ensure you have user defined that is in the MQ-admin group, and one that is not. For my testing 'mqadm' is in the MQ-admin group, 'alice' is NOT.

CHLAUTH rules:

a) Default CHLAUTH RULES in place:
- NOACCESS to all MQ-admins all channels
- NOACCESS to all SYSTEM.* channels
- allow access to SYSTEM.ADMIN.SVRCONN (non MQ-admin users..)

Due to the "NOACCESS to all MQ-admins all channels" CHLAUTH rule, by default MQ-admins can NOT access via any channels.

b) For MQ-admin access on specific channel, certain userid and certain ip address range.
Add 3 rules to allow specific user to access ADMIN.CHAN as MQ-admin
from certain ip address.
.
- Set NOACCESS from any address
- Set blockuser for this channel to only block user 'nobody' , this overrides the *MQADMIN blockuser
- ALLOW access to user 'mqadm' on specific subnet of addresses, and MAP to mqadm user authority

runmqsc:

```
SET CHLAUTH (ADMIN.CHAN) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)

SET CHLAUTH('ADMIN.CHAN') TYPE(blockuser) +
DESCR('Rule to override *MQADMIN blockuser on this channel') +
USERLIST('nobody')  ACTION(replace)

SET CHLAUTH('ADMIN.CHAN') TYPE(USERMAP) +
  CLNTUSER('mqadm')  USERSRC(MAP) MCAUSER('mqadm') +
  ADDRESS('192.168.1.*')  +
  DESCR('Allow mqadm as mqadm on local subnet') ACTION(ADD)
```

At this point, the user 'mqadm' can access/start the ADMIN.CHAN channel, from the specified ip address range.
You can run MATCH (RUNCHECK) at any time to see the results of each of these commands.

runmqsc:

```
DISPLAY CHLAUTH (ADMIN.CHAN) MATCH (RUNCHECK) CLNTUSER ('mqadm') ADDRESS
('192.168.1.138')

AMQ8878: Display channel authentication record details.
   CHLAUTH(ADMIN.CHAN)                     TYPE(USERMAP)
   ADDRESS(192.168.1.*)                  CLNTUSER(mqadm)
   MCAUSER(mqadm)
```

```
 DISPLAY CHLAUTH (ADMIN.CHAN) MATCH (RUNCHECK) CLNTUSER ('alice') ADDRESS
('192.168.1.138')

AMQ8878: Display channel authentication record details.
   CHLAUTH(ADMIN.CHAN)                     TYPE(ADDRESSMAP)
   ADDRESS(*)                              USERSRC(NOACCESS)
```

At this point, only the user's which have a CHLAUTH record would be allowed to access via the ADMIN.CHAN

Scenario 2: Control access for specific MQ client application

For this scenario, the default CHLAUTH rules are adequate, assuming MQ authority should be set for a specific user to provide the correct MQ authority (via setmqaut). For this example, let say that the authorities are set for a user 'mqapp1'.

Make a SVRCONN channel to be used for this scenario.
  APP1.CHAN - Channel to be used by particular application, specific user.

```
runmqsc:  DEFINE CHANNEL (APP1.CHAN)  CHLTYPE (SVRCONN) TRPTYPE (TCP)
```

a) With the default rules in place,
 Default CHLAUTH RULES in place:
- NOACCESS to all MQ-admins all channels
- NOACCESS to all SYSTEM.* channels
- allow access to SYSTEM.ADMIN.SVRCONN (non MQ-admin users..)

At this point, user 'mqapp1', who is NOT a MQ-admin user, can start the APP1.CHAN channel, actually any non-MQ-admin can start/access this channel. The userid coming from the MQ client application will be used for MQ object authority checking. In this case, assuming the 'mqapp1' user is running the MQ client app, this would be used for MQ object authority checking, thus if 'mqapp1' has access to the MQ objects the app needs, all will be fine, if not they will get authority errors.

You could further lock down access by creating specific CHLAUTH rules for the 'mqapp1' userid, but this may not be necessary. Note: MQ-admin access is not allowed on this channel, due to the default rule.

```
runmqsc:
SET CHLAUTH (APP1.CHAN) TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)

SET CHLAUTH('APP1.CHAN') TYPE(USERMAP) +
  CLNTUSER('mqapp1')  USERSRC(MAP) MCAUSER('mqapp1') +
  DESCR('Allow mqadm as mqadm on local subnet') ACTION(ADD)
```

Scenario 3: Control access for specific user via the user's certificate distinguished name (DN)

An example of the CHLAUTH rules to allow this case is already provided in the following knowledgecenter link:

https://www.ibm.com/support/knowledgecenter/SSFKSJ_9.2.0/com.ibm.mq.sec.doc/q132570_.htm

For this case, the user must have a certificate which is flowed to the queue manager. The DN is then matched against the SSLPEER setting of the CHLAUTH rule. The SSLPEER can use wildcards. If matched, the user can also be mapped to a different MCAUSER for purposes of checking the MQ object authorities. Mapping the MCAUSER can minimize the number of users that need to be managed in the MQ object authority manager (OAM).

Example: (SSL channel, with certificates in use)
You would like rules to:
1) block all users for particular channel
2) allow only users with particular SSLPEER and use the user's client user for MQ OAM access.

```
.
# block all users on any IP address.
SET CHLAUTH('SSL1.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('*')
USERSRC(NOACCESS) DESCR("block all") WARN(NO) ACTION(ADD)
.
# override - no MQM admin rule (allow mqm group /mqm admin users to
connect.
SET CHLAUTH('SSL1.SVRCONN') TYPE(BLOCKUSER) USERLIST('nobody')
DESCR('override no mqm admin rule') WARN(NO) ACTION(ADD)
.
# allow particular SSLPEER, use client id coming in from channel
SET CHLAUTH('SSL1.SVRCONN') TYPE(SSLPEERMAP)
SSLPEER('CN=MCREGGE,O=IBM,C=US') USERSRC(CHANNEL) ACTION(ADD)
```

The client userid connecting on the channel will be used for the MQ OAM authority of MQ objects; thus the userid must have appropriate MQ authorities. You could map to a different MQ userid if you wanted to, by using:
USERSRC(MAP) MCAUSER('mquser1') instead of USERSRC(CHANNEL).

Scenario 4: Mapping a particular user to the mqm user (extension of scenario 1)

This is an addition/modification to scenario 1 above. Add the following CHLAUTH rule to map particular users to the mqm user, or an MQ-admin userid, which has MQ object authority setup in the MQ OAM.

runmqsc:

```
SET CHLAUTH('ADMIN.CHAN') TYPE(USERMAP) +
  CLNTUSER ('mcregge')  USERSRC(MAP) MCAUSER ('mqm') +
  ADDRESS('192.168.1-100.*')  +
  DESCR ('Allow mcregge as MQ-admin on local subnet') ACTION (ADD)
```

This will allow and map the 'mcregge' user over to the 'mqm' user for the particular channel ADMIN.CHAN.

Scenario 5: Only allow access to a particular channel from a specific IP address range.

- set No access to the channel from anywhere
- Allow access from a specific ip address or address range

```
SET CHLAUTH('APP2.CHAN') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
WARN(NO) ACTION(ADD)

SET CHLAUTH('APP2.CHAN') TYPE(ADDRESSMAP) ADDRESS('9.95.100.1-5') USERSRC(MAP)
MCAUSER('mqapp2') ACTION(ADD)
```

This will only allow the APP2.CHAN channel to be started when the connection comes from the specific ip address range specified. The user connecting MCAUSER will be mapped to 'mqapp2' and thus get the MQ OAM authority for that user.

Scenario 6: For a specific channel, Block all users, but allow specific users to connect.

For access on channel 'MY.SVRCONN':

a) Default CHLAUTH RULES in place:
- NOACCESS to all MQ-admins all channels
- NOACCESS to all SYSTEM.* channels
  - allow access to SYSTEM.ADMIN.SVRCONN (non MQ-admin users..)
  -

b) then add the following:

```
# block all users
SET CHLAUTH('MY.SVRCONN') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS) DE-
SCR("block all") WARN(NO) ACTION(ADD)

# override - no MQM admin rule
SET CHLAUTH('MY.SVRCONN') TYPE(BLOCKUSER) USERLIST('nobody') DESCR('override no
mqm admin rule') WARN(NO) ACTION(ADD)

# allow mcregge userid
SET CHLAUTH('MY.SVRCONN') TYPE(USERMAP) CLNTUSER('mcregge')
  USERSRC(CHANNEL) DESCR('allow mcregge userid') ACTION(ADD)
```

This first blocks anyone from connecting on MY.SVRCONN, then it will only allow the MY.SVRCONN channel to be started when the connection comes from the specific userid 'mcregge'. The user connecting on the channel ('mcregge') will be used for the MQ OAM authority of MQ objects; thus the userid must have appropriate MQ authorities. You could map to a different MQ userid if you wanted to, by using USERSRC(MAP) MCAUSER('mquser1') instead of USERSRC(CHANNEL).

Scenario 7: Using CHLAUTH for RCVR (Receiver/Sender) channels:

You can use CHLAUTH rules to add extra security to Sender/Receiver channels, to lock down who can connect to the Receiver channel.

CHLAUTH rules can be used on any channel, but there are some restrictions. USERMAP rules are only for SVRCONN channels

Additionally, if adding or making changes to CHLAUTH rules, the updated CHLAUTH rules only apply when starting the channel, so if channels are already running, you would need to stop and restart them, for the CHLAUTH updates to apply.

To lock down a Sender/Receiver channel pair, you could add CHLAUTH rules on the RCVR side to determine who can start the channel. Here are some examples:

This example only allows a connection from a particular ipaddress to start the 'TO.MYSVR1' channel:

First you could lock down the channel by disallowing all
for channel 'TO.MYSVR1', RCVR channel

```
SET CHLAUTH('TO.MYSVR1') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
DESCR('Back-stop rule')
```

Then you could allow this channel to be started

```
SET CHLAUTH('TO.MYSVR1') TYPE(ADDRESSMAP) ADDRESS('192.168.1.134') USERSRC(MAP)
MCAUSER('mqapp') ACTION(ADD)
```

Another example would be to only allow the connection from a particular queue manager:

Lock down all access:

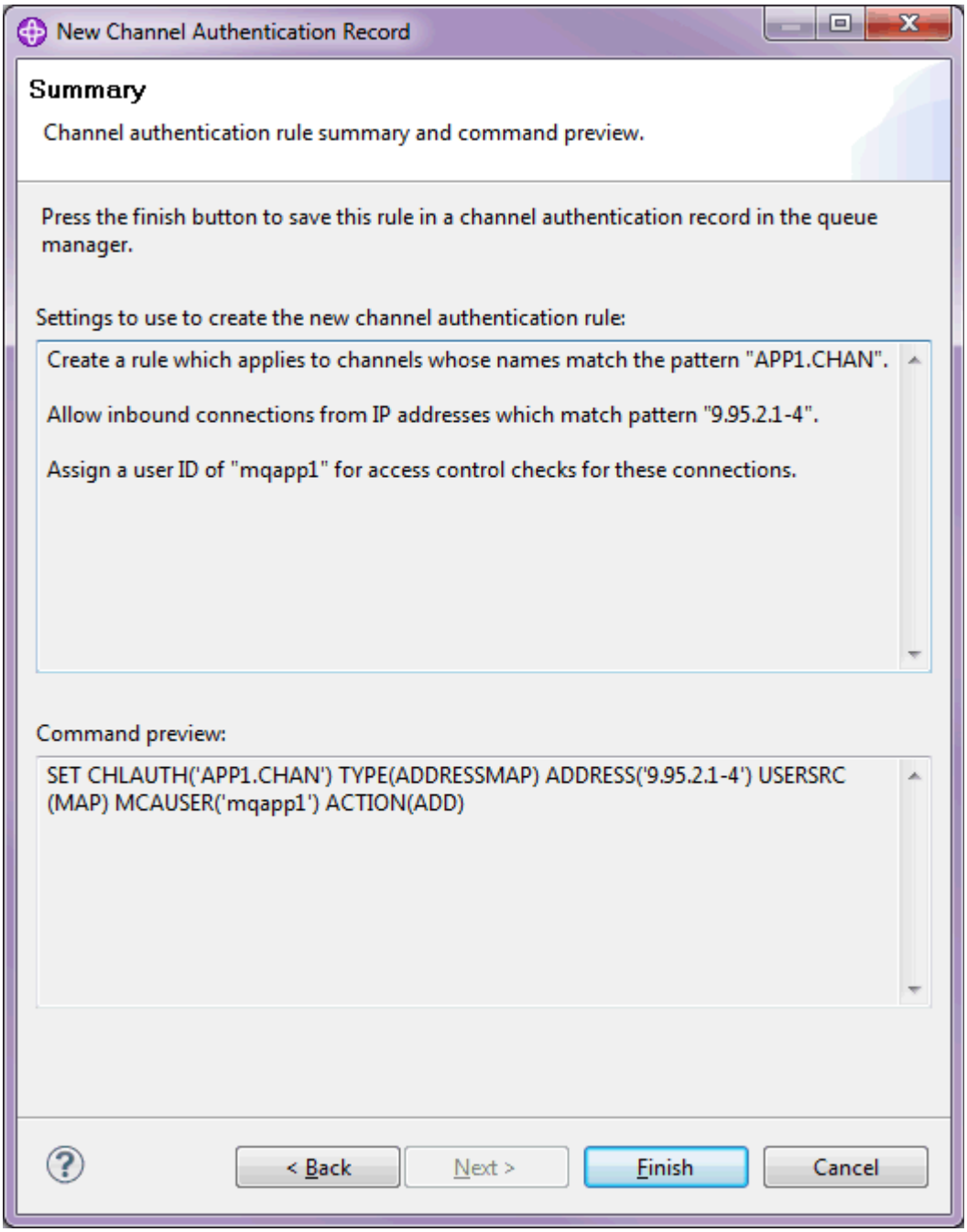
```
SET CHLAUTH('TO.MYSVR1') TYPE(ADDRESSMAP) ADDRESS('*') USERSRC(NOACCESS)
DESCR('Back-stop rule')
```

Then allow access from queue manager MYSVR2 and from a particular ipaddress:

```
SET CHLAUTH('TO.MYSVR1') TYPE(QMGRMAP) QMNAME('MYSVR2') USERSRC(MAP)
MCAUSER('mqapp') ADDRESS('192.168.1.134') ACTION(ADD)
```

MQ Explorer: Wizard to create CHLAUTH rules

If you right-click on the Channel Authentication Records option in the MQ Explorer navigator under Channels for a queue manager, MQ Explorer has an excellent wizard which steps you through building your CHLAUTH rules. On the last screen, it actually presents you with a summary of the CHLAUTH rule and the command used to create the CHLAUTH rule.



Summary:

I hope this information and examples provide a better understanding of using the CHLAUTH rules. If you provide feedback for other scenarios you would like to see, we can try to update this document with more examples.

Additional Resources:

See the Channel Authentication in the knowledge center at:
http://www.ibm.com/support/knowledgecenter/SSFKSJ_8.0.0/com.ibm.mq.sec.doc/q010250_.htm

See the CHLAUTH command options in our infocenter:
https://www.ibm.com/support/knowledgecenter/SSFKSJ_9.2.0/com.ibm.mq.ref.adm.doc/q086630_.htm

Presentation/Recording: WebSphere MQ: Using CHLAUTH to lock down Administrative access with MQ Explorer
http://www.ibm.com/support/docview.wss?uid=swg27039600

Redbook: Secure Messaging with WMQ:
http://www.redbooks.ibm.com/abstracts/sg248069.html

CHLAUTH feature in WebSphere MQ v7.1:
https://www.ibm.com/developerworks/community/blogs/aimsupport/entry/chlauth_feature_in_mq_71?lang=en

CHLAUTH – the back-stop rule:
https://www.ibm.com/support/pages/chlauth-back-stop-rule-content-author-morag-hughson

I'm being blocked by CHLAUTH - how can I work out why?
https://mqgem.wordpress.com/2013/02/09/blocked_by_chlauth_why/

WMQ 7.1/7.5 queue manager RC 2035 / AMQ4036:
http://www.ibm.com/support/docview.wss?uid=swg21577137

How to remove a CHLAUTH record:
http://www.ibm.com/support/docview.wss?uid=swg21577138