



IBM Software Group

MQ Pub/Sub: topic tree, security

<http://www.ibm.com/support/docview.wss?uid=swg27050162>

Angel Rivera (rivera@us.ibm.com)

IBM MQ Distributed Level 2 Support

Date last updated: 20-Sep-2017

Link to index: <https://developer.ibm.com/answers/questions/402074/mq-pubsub-training-presentations.html>



ON DEMAND BUSINESS™

Agenda

- This presentation is a continuation of:
 - <http://www.ibm.com/support/docview.wss?uid=swg27050138>
 - MQ Pub/Sub: non-durable topics and subscribers
- Security – authorities needed for non-administrators
- Administration
- Topic Objects
- Topic tree



Non-admin user tries to publish/subscribe

- The host of the queue manager has user 'alice' who does not belong to group "mqm".
- Alice belongs to group 'mqusers'.
- It is a non-administrator for MQ.
- `alice@mosquito: /home/alice`
- `$ id alice`
- `uid=1008(alice) gid=1005(mqusers) groups=1005(mqusers)`

- For comparison, here is the id info for "mqm"
- `mqm@mosquito: /home/mqm`
- `$ id mqm`
- `uid=501(mqm) gid=501(mqm) groups=501(mqm)`

Non-admin user tries to publish/subscribe

- User 'alice' tries to publish into topic string 'sales'.
- The attempt fails with reason code 2035 for the action MQCONN (MQ Connection)
- The MQ client code will not indicate the root cause.

- `$ amqspub sales QMPS`
- Sample AMQSPUBA start
- MQCONN ended with reason code 2035

- `$ mqrc 2035`
- `2035 0x000007f3 MQRC_NOT_AUTHORIZED`

Details for 2035 in error log of queue mgr

- An MQ administrator (such as user 'mqm') will need to look at the recent entries of the error log for the queue manager to find out more details about the security failure.
- mqm@mosquito:
 - `$ cd /var/mqm/qmgrs/QMPS/errors`
 - `$ tail -10 AMQERR01.LOG`

Error message regarding lack of 'connect'

- Date stamp - Process(26420.12) User(mqm) Program(amqzlaa0)
- Host(mosquito) Installation(Installation5)
- VRMF(9.0.3.0) QMgr(QMPS)

- **AMQ8077: Entity 'alice' has insufficient authority to access object 'QMPS'.**

- EXPLANATION:
 - The specified entity is not authorized to access the required object. The
 - following requested permissions are unauthorized: **connect**

- ACTION:
 - Ensure that the correct level of authority has been set for this entity against
 - the required object, or ensure that the entity is a member of a privileged
 - group.



Notes: Giving authority access

notes

The userids that are not members of the MQ administrator's group "mqm" by default do not have access to the queue manager, nor other objects that are necessary for the communication between the Explorer and the remote queue manager.

Therefore, it is necessary for an MQ administrator to grant certain authorities to these users to be allowed to access the queue manager.

.

The purpose of the following setmqaut commands is:

1. GENERAL: Grant authority to access the queue manager.
2. MQ EXPLORER: Grant authority to the client channel to get the command server reply messages.
3. MQ EXPLORER: Grant authority to put messages onto the command server input queue.
4. MQ EXPLORER: Grant authority to get the reply messages.

Notes: Giving authority access

n
o
t
e
s

The queue manager name is MYQMGR.

The flag-value pair "t qmgr" refers to the object type of QueueManager.

The flag-value pair "t q" refers to the object type of Queue.

For a userid:

Issue these setmqaut commands to grant minimal authority to the userID "myuser"
(using the PrincipalName, or person, flag: -p)

```
setmqaut -m MYQMGR -t qmgr -p myuser +connect +inq +dsp
```

```
setmqaut -m MYQMGR -t q -n SYSTEM.DEFAULT.MODEL.QUEUE -p myuser +inq +browse +get +dsp
```

```
setmqaut -m MYQMGR -t q -n SYSTEM.ADMIN.COMMAND.QUEUE -p myuser +inq +put +dsp
```

```
setmqaut -m MYQMGR -t q -n SYSTEM.MQEXPLORER.REPLY.MODEL -p myuser +inq +browse +get +dsp
```

For a group:

Issue these setmqaut commands to grant minimal authority to the Unix group
(using the GroupName flag: -g)

```
setmqaut -m MYQMGR -t qmgr -g mygroup +connect +inq +dsp
```

```
setmqaut -m MYQMGR -t q -n SYSTEM.DEFAULT.MODEL.QUEUE -g mygroup +inq +browse +get +dsp
```

```
setmqaut -m MYQMGR -t q -n SYSTEM.ADMIN.COMMAND.QUEUE -g mygroup +inq +put +dsp
```

```
setmqaut -m MYQMGR -t q -n SYSTEM.MQEXPLORER.REPLY.MODEL -g mygroup +inq +browse +get +dsp
```


Non-admin user gets connect permissions

- The MQ Administrator issues the following commands to allow user alice to connect to the queue manager via line commands and MQ Explorer.

```
setmqaut -m QMPS -t qmgr -p alice +connect +inq +dsp
```

```
setmqaut -m QMPS -t q -n SYSTEM.DEFAULT.MODEL.QUEUE -p alice +inq +browse +get +dsp
```

```
setmqaut -m QMPS -t q -n SYSTEM.ADMIN.COMMAND.QUEUE -p alice +inq +put +dsp
```

```
setmqaut -m QMPS -t q -n SYSTEM.MQEXPLORER.REPLY.MODEL -p alice +inq +browse +get +dsp
```



Non-admin user tries to publish/subscribe

- User 'alice' tries again.
 - The attempts fails again with rc 2035.
 - Notice that the rc is for MQOPEN (open a topic), not MQCONN.
-
- \$ amqspub sales QMPS
 - Sample AMQSPUBA start
 - target topic is sales
 - MQOPEN ended with reason code 2035
 - unable to open topic for publish
 - Sample AMQSPUBA end

Non-admin user tries to publish/subscribe

- Again, the MQ admin needs to look at the bottom of the error log of the queue manager.
- Notice that now the permission is for “pub” (previously it was for “connect”).
- AMQ8009: Entity 'alice' has insufficient authority to access **topic string 'sales'**.
- EXPLANATION:
 - The specified entity is not authorized to access the required topic.
 - The following permissions were requested: **pub**

Notes: Giving authority access

n
o
t
e
s

<http://www-01.ibm.com/support/docview.wss?uid=swg27016153>

Authorizations needed for non-mqm users to publish and subscribe to Topics in MQ V7

.
+++ Summary

.
To let the users in the groups "editors" and "journalists" to connect to the queue manager WMQ7:

```
setmqaut -m WMQ7 -t qmgr -g editors +connect +inq +dsp
```

```
setmqaut -m WMQ7 -t qmgr -g journalists +connect +inq +dsp
```

.
To let the users in the group "editors" subscribe to topic object "DELI" on Queue Manager WMQ7 and to resume durable subscriptions:

```
setmqaut -m WMQ7 -n DELI -t topic -g editors +sub +resume
```

.
To allow users in the group "journalists" to publish to the topic:

```
setmqaut -m WMQ7 -n DELI -t topic -g journalists +pub
```

Specify authority on durable Topic Object

- It is NOT possible to specify authority permissions on **topic strings**.
- In this case, at this time, the topic 'sales' is just a topic string and it is not durable.
- It is necessary to specify the authority on **a Topic Object (which is a durable topic)**

Notes: Default set of durable topic objects

n
o
t
e
s

display topic(*)

AMQ8633I: Display topic details.

TOPIC(SYSTEM.ADMIN.TOPIC) TYPE(LOCAL)

AMQ8633I: Display topic details.

TOPIC(SYSTEM.BASE.TOPIC) TYPE(LOCAL)

AMQ8633I: Display topic details.

TOPIC(SYSTEM.BROKER.ADMIN.STREAM) TYPE(LOCAL)

AMQ8633I: Display topic details.

TOPIC(SYSTEM.BROKER.DEFAULT.STREAM) TYPE(LOCAL)

AMQ8633I: Display topic details.

TOPIC(SYSTEM.BROKER.DEFAULT.SUBPOINT) TYPE(LOCAL)

AMQ8633I: Display topic details.

TOPIC(SYSTEM.DEFAULT.TOPIC) TYPE(LOCAL)

Notice: The topic **SYSTEM.BASE.TOPIC** is a key durable topic object ...
... but for security reasons, do NOT give authority for this topic !!!

Assigning authorities for Pub/Sub

- Question 1:
- Should authorities be given for the SYSTEM.BASE.TOPIC?

- Answer:
- **No!!!!**
- Even though giving permissions to user 'alice' to do publishing actions against the SYSTEM.BASE.TOPIC would resolve the short term security issue...
- ... it is NOT a Best Practice to do so, because it will give too much power to this non-administrator for tasks related to Pub/Sub

Assigning authorities for Pub/Sub

- Question 2:
- If authorities should NOT be given for the `SYSTEM.BASE.TOPIC`, then what else can be done to allow user 'alice' to publish into a topic?
- Answer:
- It is necessary to create a durable topic.
- It is also known as a “Topic Object” and will survive a restart of the queue manager.
- A ‘Topic Object’ contains an attribute that is a “topic string”



Defining a Topic Object

- To define a topic object called 'SALES' that has a topic string of 'sales', the MQ administrator issues:

- **define topic('SALES') topicstr('sales')**

- **display topic('SALES')**

- AMQ8633I: Display topic details.

■ <u>TOPIC (SALES)</u>	TYPE (LOCAL)
■ <u>TOPICSTR (sales)</u>	DESCR ()
■ CLUSTER ()	CLROUTE (DIRECT)
■ DURSUB (ASPARENT)	PUB (ASPARENT)
■ SUB (ASPARENT)	DEFPSIST (ASPARENT)
■ DEFPRTY (ASPARENT)	DEFPRESP (ASPARENT)
■ PMSGDLV (ASPARENT)	NPMSGDLV (ASPARENT)
■ PUBSCOPE (ASPARENT)	SUBSCOPE (ASPARENT)
■ PROXYSUB (FIRSTUSE)	WILDCARD (PASSTHRU)
■ MDURMDL ()	MNDURMDL ()
■ MCAST (ASPARENT)	COMMINFO ()
■ USEDLO (ASPARENT)	CUSTOM ()

Assigning authorities to a topic object

- Now that there is a durable topic, the MQ Administrator issues the following command to allow user 'alice' to perform the action “pub” on the topic object 'SALES' which has a topic string “sales”.
- Thus, the authorities for a topic string are done indirectly via a Topic Object.
- `mqm@mosquito: /home/mqm`
- `$ setmqaut -m QMPS -n 'SALES' -t topic -p alice +pub`

Non-admin user 'alice' tries to publish

- User 'alice' tries again.
- This time the attempt is successful!

- `alice@mosquito: /home/alice`
- `$ amqspub sales QMPS`
- Sample AMQSPUBA start
- target topic is sales
- **This is a test from Alice**
- Sample AMQSPUBA end



Let's explore the topic object 'SALES'

- Showing some attributes for the topic:

- display topic(SALES)

- AMQ8633I: Display topic details.

▪ TOPIC (SALES)	TYPE (LOCAL)
▪ TOPICSTR (sales)	DESCR ()
▪ CLUSTER ()	CLROUTE (DIRECT)
▪ DURSUB (ASPARENT)	PUB (ASPARENT)
▪ SUB (ASPARENT)	DEFPSIST (ASPARENT)

- What is the meaning of the value 'ASPARENT'?

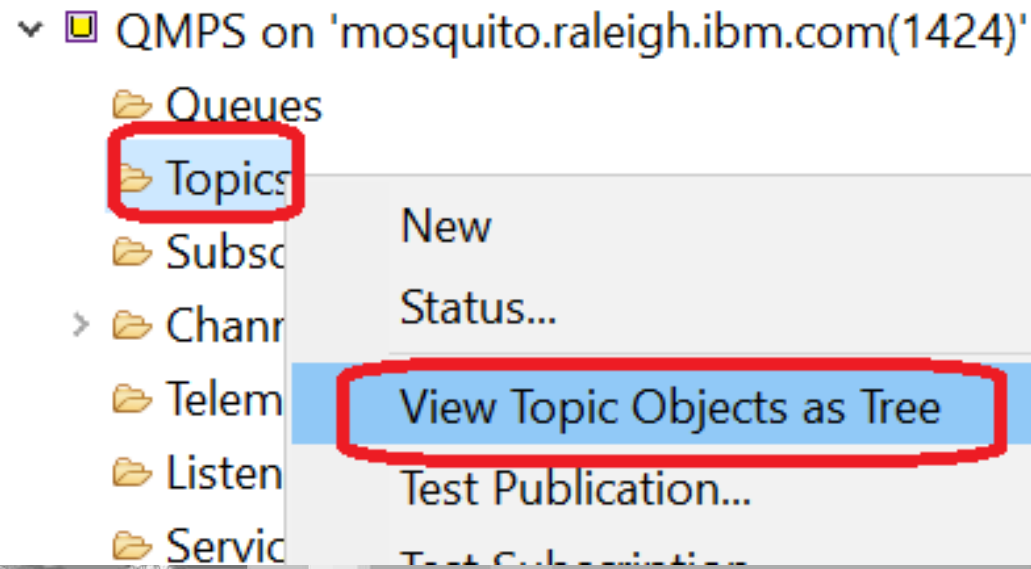
- It means "**As Parent**": it refers to the value of the attribute of the PARENT of this topic object.

This implies a hierarchy!



Viewing Topic Objects as Tree

- From MQ Explorer, select “Topics”, right click and then select:
- View Topic Objects as Tree



Viewing Topic Objects as Tree

- You will see a new panel: MQ Topic Viewer
 - Move the cursor on top of the topic string 'sales'
- Notice a yellow box that shows the Topic Object

MQ Topic Viewer

Topic Objects on QMPS

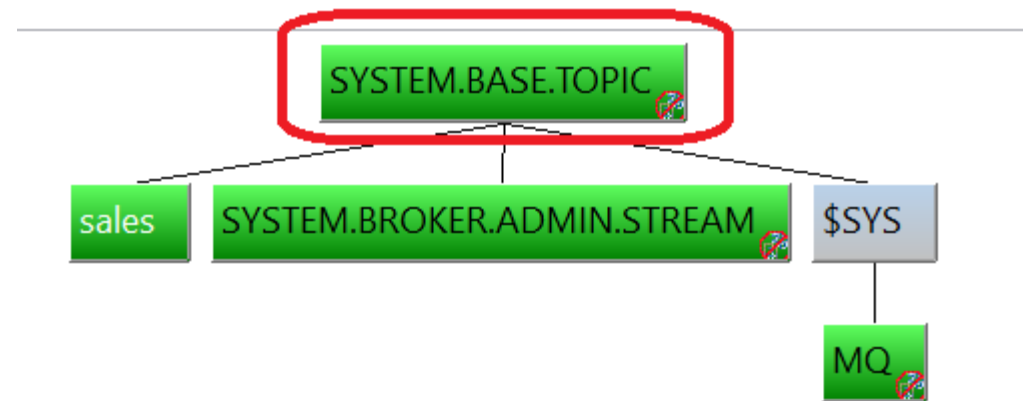
```
graph TD
    A[SYSTEM.BASE.TOPIC] --- B[sales]
    A --- C[SYSTEM.BROKER.ADMIN.STREAM]
    A --- D[$SYS]
    D --- E[MQ]
```

Object Name:	SALES
Topic String:	sales
Description:	

SYSTEM.BASE.TOPIC (root of topic tree)

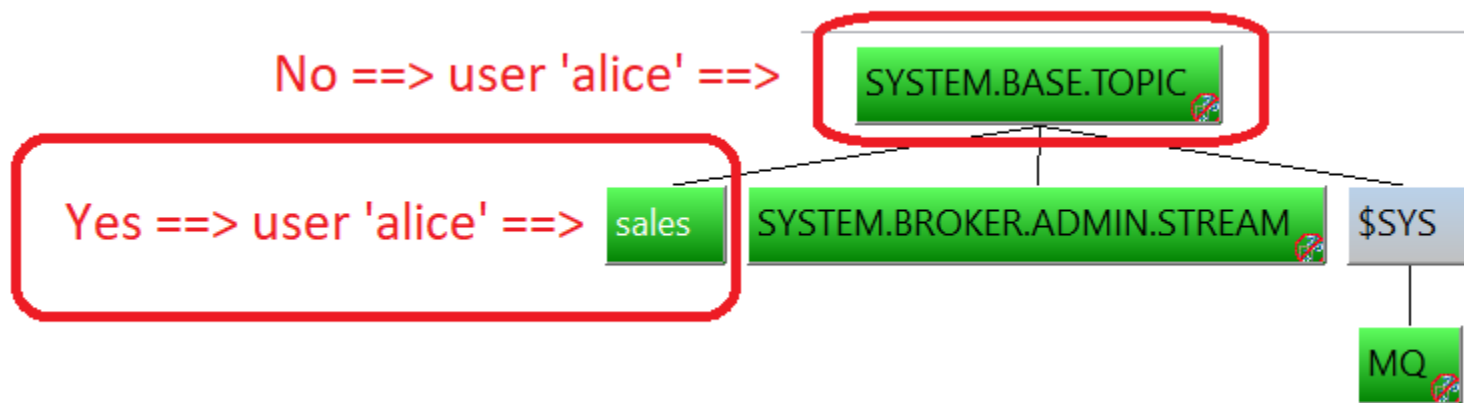
- Notice that the root of the Topic Tree is:
SYSTEM.BASE.TOPIC

All attributes from this Topic Object are inherited to the rest of the topic objects, including SECURITY!



Give security at child topic objects

- Thus, do NOT give security access to non-admins at this topic object, because you will give them too much power.
- Give the access to a child topic object such 'SALES' (associated to topic string 'sales')



Value 'ASPARENT' (As Parent)

- For 'SALES', the value of DEFPSIST (default persistence) is ASPARENT
 - display topic(SALES) defpsist
 - AMQ8633I: Display topic details.
 - TOPIC(SALES) TYPE(LOCAL)
 - DEFPSIST(**ASPARENT**)

- Let's take a look at the parent:
 - display topic(SYSTEM.BASE.TOPIC) defpsist
 - TOPIC(SYSTEM.BASE.TOPIC) TYPE(LOCAL)
 - DEFPSIST(**NO**)

- Thus, the effective value for topic SALES is: NO

Value 'ASPARENT' (As Parent)

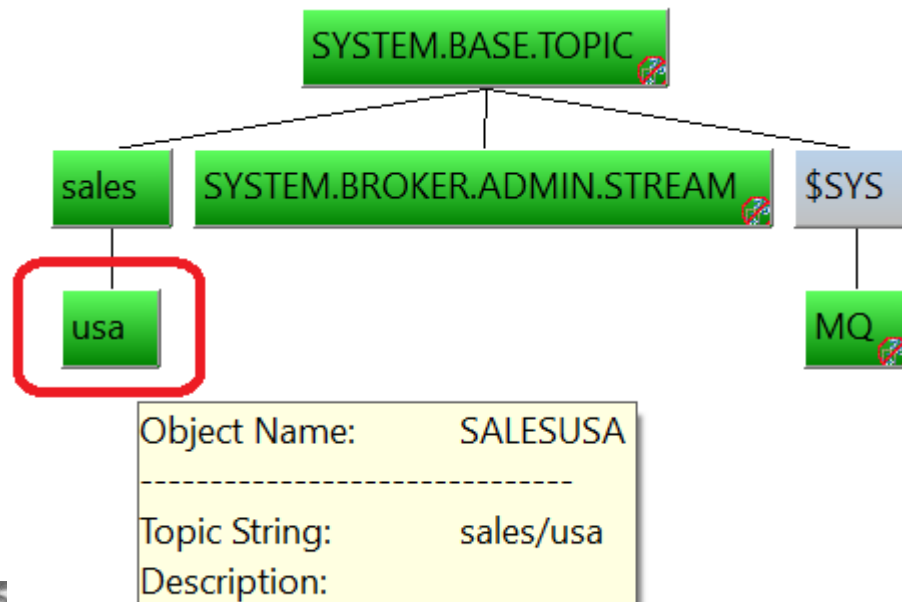
- If the value for DEFPSIST for SYSTEM.BASE.TOPIC is changed from NO to YES
- then effective value for 'SALES' will be YES

- For a Child topic object, if the value of DEFPSIST is not ASPARENT, then the change of the value at the Parent topic object will not affect the value of the Child topic object.



Adding a child of a child in the topic tree

- Let's add a new topic object.
- The / is a delimiter inside the topic string.
- define topic('SALESUSA') topicstr('sales/usa')
- Viewing the updated topic tree:



MQ samples use topic strings

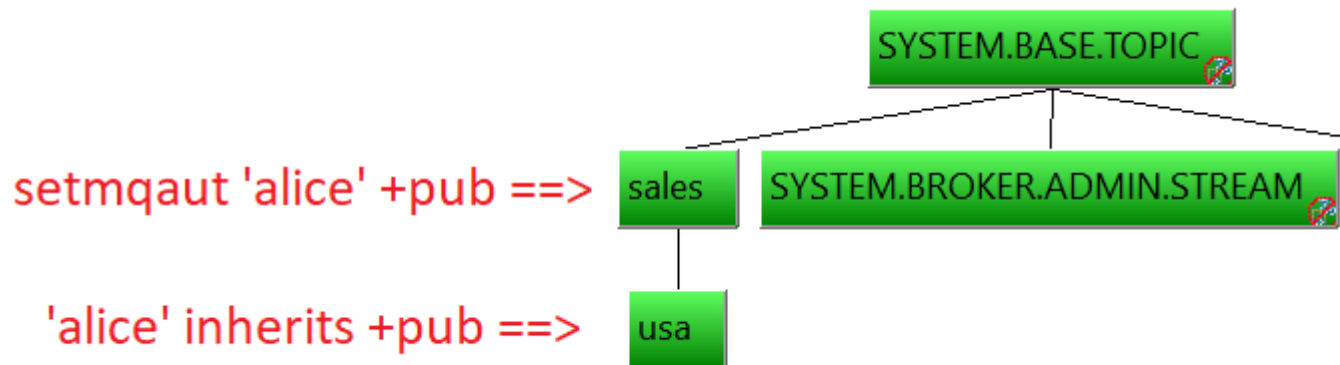
- The MQ samples work on the **topic strings** and
- not on the topic objects
- The following commands will fail for user 'alice' with 2035, because the sample will treat 'SALES' as a topic string, and there is no topic object that has a topic string of 'SALES'.
 - `alice@mosquito: /home/alice`
 - `$ amqspub SALES QMPS`
 - Sample AMQSPUBA start
 - target topic is SALES
 - MQOPEN ended with reason code 2035
 - unable to open topic for publish
 - Error: AMQ8009W: Entity 'alice' has insufficient authority to access topic string 'SALES'.

Publishing with compounded topic strings

- You can use a topic string that has several tokens:
 - \$ amqspub **sales/usa** QMPS
 - Sample AMQSPUBA start
 - target topic is sales/usa
 - Testing sales/usa
 - Sample AMQSPUBA end
- You can also use single quotes or double quotes
 - \$ amqspub 'sales/usa' QMPS
 - \$ amqspub "sales/usa" QMPS

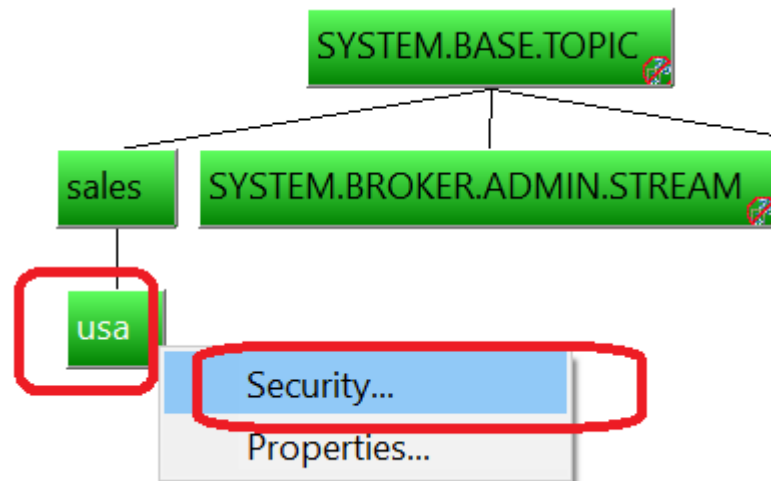
Security is inherited from Parent

- Why user 'alice' was able to publish successfully into the topic string "sales/usa", but she does not have explicit "pub" authority for "sales/usa" (setmqaut was used only for "sales")?
- Answer:
- The security is inherited.



Showing security from MQ Topic Viewer

- The MQ Topic Viewer in MQ Explorer allows you view the security settings for topics and users.
- Select the node for “usa”, right click and select “Security...”



Showing security from MQ Topic Viewer

- Specify
- User 'alice'
- Select:
- (*) User
- Click:
- Check
- Permissions

Topic Permissions for sales/usa

Name:

User Group

This panel allows you to check whether a specific user or group is allowed to publish or subscribe to the selected topic. It checks each level in the topic tree until it either receives a "Yes" answer or reaches the root.

Topic Object	Topic Component	Pub	Sub
SALESUSA	usa		
SALES	sales	✓ Y...	
SYSTEM.BASE.TOPIC			
Summary		✓ Y...	✗ N..

Notes: Security is based in groups

n

o

t

e

s

- The dialog from the previous slide has a checkbox for “Users” and the code for the dialog handles the inquiries related to authority, based on the group membership for the user.
- However, if you are going to query the queue manager via runmqsc, you will not find references to the user ‘alice’, but rather, to the groups to which this user belongs, in this case the group is “mqusers”
- `alice@mosquito: /home/alice`
- `$ id alice`
- `uid=1008(alice) gid=1005(mqusers) groups=1005(mqusers)`
- The MQ runmqgras utility includes the information about authorities in 2 files:
 - `amqoamd_QMPS.stdout`
 - `dmpmqaut_QMPS.stderr` (note that `dmpmqaut_QMPS.stdout` is always empty!)
- And if you specify “-section defs” then you will get a 3rd file with information:
 - `defs/runmqsc_AUTHREC_QMPS.stdout`

Notes: Security is based in groups

notes

- Let's explore the relevant contents of these 3 files for the group "mqusers"
- The command used was: **dmpmqaut -m QMPS > dmpmqaut_QMPS.stderr 2>&1**
- + File: dmpmqaut_QMPS.stderr
- profile: SALES
- object type: topic
- entity: mqusers
- entity type: group
- authority: pub

- The command used was: **runmqras -qmlist QMPS -section defs**
- + File: defs/runmqsc_AUTHREC_QMPS.stdout
- profile: SALES
- object type: topic
- entity: mqusers
- entity type: group
- authority: pub

Notes: Security is based in groups

notes

- The command used was: **amqoamd -m QMPS > amqoamd_QMPS.stdout**
- + File: amqoamd_QMPS.stdout
- ObjectType: topic
- ObjectName: SALES
 - Principal: mqm
 - Authorities: passall passid setall setid chg clr dlt dsp ctrl pub sub resume (0x027e3fff)
- **Principal: mqusers**
- **Authorities: pub (0x00000800)**

- Note that this file includes for each object, the different entities that have authorities.
- Thus, be careful to select the proper rows for the desired group!
- A common mistake is to just look at the top rows for group “mqm”!

Notes: Getting output in setmqaut format

notes

- If you want to obtain the output in the format that is used by “setmqaut”, use the following MQ utility to display the configuration of the queue manager: dmpmqcfg
- **dmpmqcfg -m QMPS -o setmqaut > QMPS.dmpmqcfg.setmqaut.sh**
- EXAMPLE OUTPUT (highlighting at the bottom the one for topic SALES)
- #####
- # Script generated on 2017-08-15 at 03.43.39
- # Script generated by user 'mqm' on host 'mosquito'
- # Queue manager name: QMPS
- # Queue manager platform: UNIX
- # Queue manager command level: (903/903) XRCAP(YES) AMQPCAP(YES)
- # Command issued: dmpmqcfg -m QMPS -o setmqaut
- #####
- setmqaut -m QMPS -n SYSTEM.BROKER.INTER.BROKER.COMMUNICATIONS -t queue -g mqm +allmqi +chg +clr +dlt +dsp
- setmqaut -m QMPS -n SALES -t topic -g mqm +chg +clr +dlt +dsp +passall +passid +setall +setid +ctrl +pub +sub +resume
- **setmqaut -m QMPS -n SALES -t topic -g mqusers +pub**

Showing security from MQ Topic Viewer

The dialog has an explanatory sentence:

“It checks each level in the topic tree until it either receives a "Yes" answer or reaches the root”

The traversal of the tree is from the leaves to root.

The first node to be checked is:

Topic Object: SALESUSA (usa)

Topic Object	Topic Component	Pub	Sub
SALESUSA	usa		

At this point, user 'alice' does NOT have +pub permission, thus, continue traversal 1 node up.

Showing security from MQ Topic Viewer

The next node is:

Topic Object: SALES (sales)

Topic Object	Topic Component	Pub	Sub
SALESUSA	usa		
SALES	sales	✓ Y...	

At this point, user 'alice' has +pub permission, thus, the traversal ends here.

Showing security from MQ Topic Viewer

Just for completeness, let's change our focus to the "sub" permissions for 'alice'.

1: This user does not have +sub permissions at SALESUSA, thus traverse 1 node up.

2: This user does not have it at SALES. Traverse.

3: This user does not have it at root. Stop.

Topic Object	Topic Component	Pub	Sub
SALESUSA	usa		
SALES	sales	✓ Y...	
SYSTEM.BASE.TOPIC			
Summary		✓ Y...	✗ N..

Showing security from MQ Topic Viewer

Let's verify it:

User 'alice' tries to subscribe to sales/usa
Attempt fails, not authorized.

```
alice@mosquito: /home/alice
```

```
$ amqssub sales/usa QMPS
```

```
Sample AMQSSUBA start
```

```
MQSUB ended with reason code 2035
```

```
unable to subscribe to topic
```

```
Sample AMQSSUBA end
```



Non-admin user 'fulano' tries to subscribe

- Another user, 'fulano' belongs to a different group 'mqviewer'.
- Does not have +sub access on any topic object.

- fulano@mosquito: /home/fulano
- \$ id fulano
- uid=1021(fulano) gid=1006(mqviewer) groups=1006(mqviewer)

- \$ amqssub sales QMPS
- Sample AMQSSUBA start
- MQCONN ended with reason code 2035

Which Topic Object to give Pub/Sub access

- Well, we know what happens next ...
- The “connect” access permissions are given and
- user ‘fulano’ tries again and will fail with rc 2035
- because this user does not have +sub access.

- Question:
- When doing setmqaut to give +sub access to ‘fulano’, which topic object needs to be specified?
- Answer:
- It depends: SALES or SALESUSA
- (But NOT SYSTEM.BASE.TOPIC !!)

setmqaut +sub

- The MQ Administrator decides that user 'fulano' should have +sub authority ...
ONLY for the sales in the USA, ...
but not for the sales for other countries.
- Notes:
 - 1: Remember to use a Topic Object name.
 - 2: Even though +resume will not be explored in this presentation, it is a best practice to specify it.
- `setmqaut -m QMPS -n 'SALESUSA' -t topic -p fulano +sub +resume`

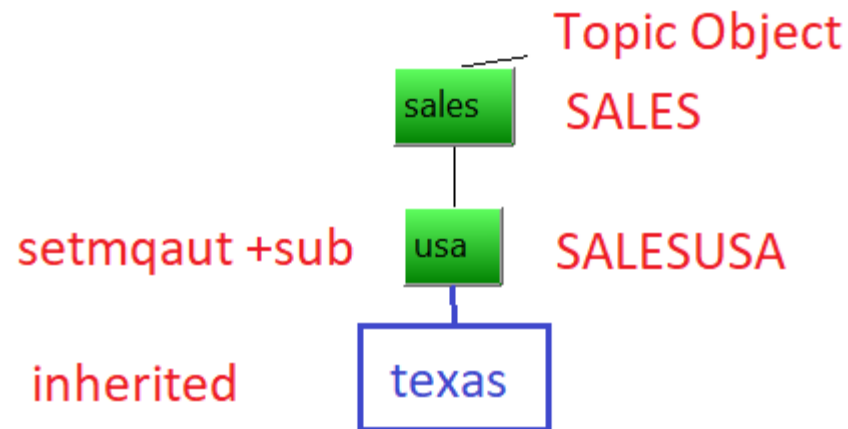
Non-admin user 'fulano' tries to subscribe

- Now user 'fulano' can subscribe to 'sales/usa':
 - fulano@mosquito: /home/fulano
 - \$ amqssub **sales/usa** QMPS
 - Sample AMQSSUBA start
 - Calling MQGET : 30 seconds wait time
- BUT cannot subscribe to other countries (rc 2035)
 - \$ amqssub **sales** QMPS
 - \$ amqssub **sales/mexico** QMPS

Non-admin user 'fulano' tries to subscribe

- However, this user can subscribe to a USA state:
- 'texas':
- \$ amqssub **sales/usa/texas** QMPS
- Sample AMQSSUBA start
- Calling MQGET : 30 seconds wait time

- Why?
- Because
- 'texas' inherits
- from 'usa'



Topic Status view – shows all topics

- Folder Topics > Status ...

Shows all topics (durable and non durable)

- In this example, 'texas' is non durable (there is no topic object for it)

 QMPS - Topic Status

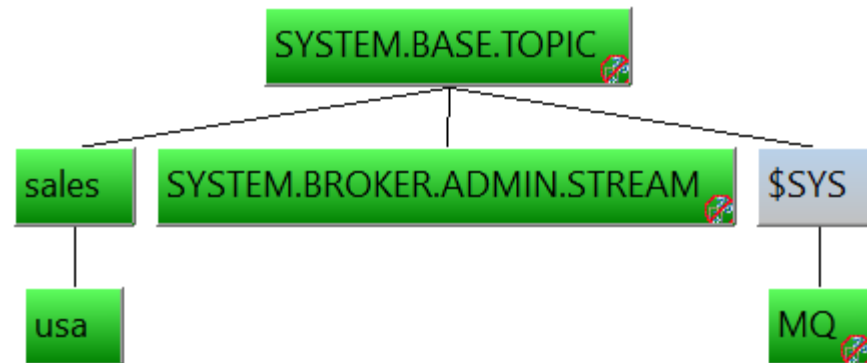
Queue Manager: QMPS

Topic status:

Topic string	Admin topic name	Sub count	Pub count
> [Empty]	SYSTEM.BASE.TOPIC	0	0
> \$SYS		0	0
▼ sales	SALES	0	0
▼ usa	SALESUSA	0	0
texas		0	0

MQ Topic Viewer – shows Topic Objects

- Folder Topics > View Topic Objects as Tree
- Shows ONLY the Topic Objects (the durable topics)
- In this example, 'texas' is non durable and therefore it is non shown in this view.



Notes: Reviewing related files from runmqras

notes

- The runmqras command can be used to get configuration information for a queue manager:
 - **runmqras -qmlist QMPS -section defs**
- In particular for Pub/Sub here are some files of interest:
 - defs/runmqsc_AUTHREC_QMPS.stdout => For security settings
 - => Already discussed in a previous slide
 - defs/runmqsc_TOPIC_QMPS.stdout => DIS TOPIC(*) ALL
 - defs/runmqsc_TPSTATUS_TOPIC_QMPS.stdout => Topic Status
 - => DIS TPSTATUS('#) TYPE(TOPIC) ALL

Notes: Display all Topic Objects

n
o
t
e
s

▪ defs/runmqsc_TOPIC_QMPS.stdout => DIS TOPIC(*) ALL

▪ AMQ8633I: Display topic details.

▪ **TOPIC(SALES)**

TYPE(LOCAL)

▪ **TOPICSTR(sales)**

DESCR()

- | | |
|----------------------|--------------------|
| ▪ CLUSTER() | CLROUTE(DIRECT) |
| ▪ DURSUB(ASPARENT) | PUB(ASPARENT) |
| ▪ SUB(ASPARENT) | DEFPSIST(ASPARENT) |
| ▪ DEFPRTY(ASPARENT) | DEFPRESP(ASPARENT) |
| ▪ PMSGDLV(ASPARENT) | NPMSGDLV(ASPARENT) |
| ▪ PUBSCOPE(ASPARENT) | SUBSCOPE(ASPARENT) |
| ▪ PROXYSUB(FIRSTUSE) | WILDCARD(PASSTHRU) |
| ▪ MDURMDL() | MNDURMDL() |
| ▪ MCAST(ASPARENT) | COMMINFO() |
| ▪ USEDLQ(ASPARENT) | CUSTOM() |

▪ AMQ8633I: Display topic details.

▪ **TOPIC(SALESUSA)**

TYPE(LOCAL)

▪ **TOPICSTR(sales/usa)**

DESCR()

- | | |
|----------------------|--------------------|
| ▪ CLUSTER() | CLROUTE(DIRECT) |
| ▪ DURSUB(ASPARENT) | PUB(ASPARENT) |
| ▪ SUB(ASPARENT) | DEFPSIST(ASPARENT) |
| ▪ DEFPRTY(ASPARENT) | DEFPRESP(ASPARENT) |
| ▪ PMSGDLV(ASPARENT) | NPMSGDLV(ASPARENT) |
| ▪ PUBSCOPE(ASPARENT) | SUBSCOPE(ASPARENT) |
| ▪ PROXYSUB(FIRSTUSE) | WILDCARD(PASSTHRU) |
| ▪ MDURMDL() | MNDURMDL() |
| ▪ MCAST(ASPARENT) | COMMINFO() |
| ▪ USEDLQ(ASPARENT) | CUSTOM() |

Notes: Display Topic Status (all topics)

notes

- defs/runmqsc_TPSTATUS_TOPIC_QMPS.stdout => Topic Status
- => DIS TPSTATUS('#') TYPE(TOPIC) ALL

- AMQ8754I: Display topic status details.
- **TOPICSTR()** **ADMIN(SYSTEM.BASE.TOPIC)**
- CLUSTER()
- COMMINFO(SYSTEM.DEFAULT.COMMINFO.MULTICAST)
- MDURMDL(SYSTEM.DURABLE.MODEL.QUEUE)
- MNDURMDL(SYSTEM.NDURABLE.MODEL.QUEUE)
- CLROUTE(NONE) DEFPSIST(NO)
- DEFPRTY(0) DEFPRESP(SYNC)
- DURSUB(YES) PUB(ENABLED)
- SUB(ENABLED) PMSGDLV(ALLDUR)
- NPMSGDLV(ALLAVAIL) RETAINED(NO)
- MCAST(DISABLED) PUBCOUNT(0)
- SUBCOUNT(0) PUBSCOPE(ALL)
- SUBSCOPE(ALL) USEDQLQ(YES)

Notes: Display Topic Status (all topics)

n
o
t
e
s

▪ TOPICSTR(sales)

- CLUSTER()
- COMMINFO(SYSTEM.DEFAULT.COMMINFO.MULTICAST)
- MDURMDL(SYSTEM.DURABLE.MODEL.QUEUE)
- MNDURMDL(SYSTEM.NDURABLE.MODEL.QUEUE)
- CLROUTE(NONE) DEFPSIST(NO)
- DEFPRTY(0) DEFPRESP(SYNC)
- DURSUB(YES) PUB(ENABLED)
- SUB(ENABLED) PMSGDLV(ALLDUR)
- NPMSGDLV(ALLAVAIL) RETAINED(NO)
- MCAST(DISABLED) PUBCOUNT(0)
- SUBCOUNT(0) PUBSCOPE(ALL)
- SUBSCOPE(ALL) USEDLQ(YES)

ADMIN(SALES)

▪ TOPICSTR(sales/usa)

- CLUSTER()
- COMMINFO(SYSTEM.DEFAULT.COMMINFO.MULTICAST)
- MDURMDL(SYSTEM.DURABLE.MODEL.QUEUE)
- MNDURMDL(SYSTEM.NDURABLE.MODEL.QUEUE)
- CLROUTE(NONE) DEFPSIST(NO)
- DEFPRTY(0) DEFPRESP(SYNC)
- DURSUB(YES) PUB(ENABLED)
- SUB(ENABLED) PMSGDLV(ALLDUR)
- NPMSGDLV(ALLAVAIL) RETAINED(NO)
- MCAST(DISABLED) PUBCOUNT(0)
- SUBCOUNT(0) PUBSCOPE(ALL)
- SUBSCOPE(ALL) USEDLQ(YES)

ADMIN(SALESUSA)

Notes: Display Topic Status (all topics)

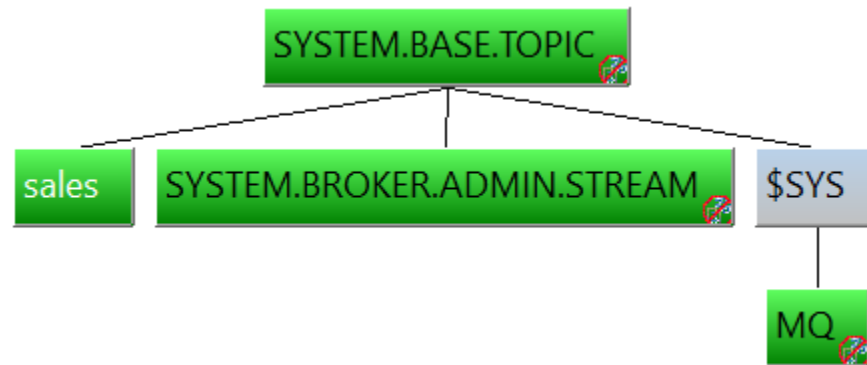
n
o
t
e
s

- Notice that this is a non-durable topic and thus the attribute ADMIN is empty (not associated to a Topic Object)

- | ▪ TOPICSTR(sales/usa/texas) | <u>ADMIN()</u> |
|---|-----------------------|
| ▪ CLUSTER() | |
| ▪ COMMINFO(SYSTEM.DEFAULT.COMMINFO.MULTICAST) | |
| ▪ MDURMDL(SYSTEM.DURABLE.MODEL.QUEUE) | |
| ▪ MNDURMDL(SYSTEM.NDURABLE.MODEL.QUEUE) | |
| ▪ CLROUTE(NONE) | DEFPSIST(NO) |
| ▪ DEFPRTY(0) | DEFPRESP(SYNC) |
| ▪ DURSUB(YES) | PUB(ENABLED) |
| ▪ SUB(ENABLED) | PMSGDLV(ALLDUR) |
| ▪ NPMMSGDLV(ALLAVAIL) | RETAINED(NO) |
| ▪ MCAST(DISABLED) | PUBCOUNT(0) |
| ▪ SUBCOUNT(0) | PUBSCOPE(ALL) |
| ▪ SUBSCOPE(ALL) | USEDLQ(YES) |

Topic tree

- Topic tree, 1 child: sales



Topic tree

- Topic tree, 1 grand-child: sales/usa

