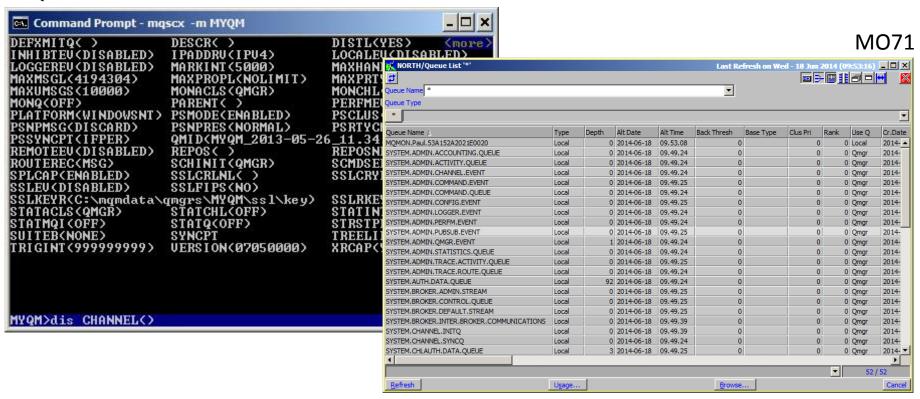
## **MQGem Administration Tools**

#### **MQSCX**





Paul Clarke

support@mqgem.com www.mqgem.com

### Agenda

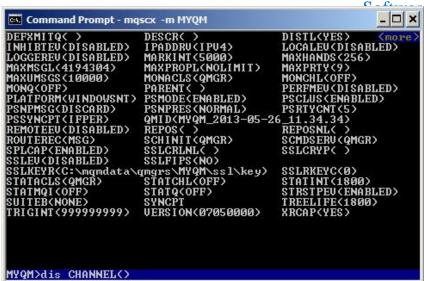


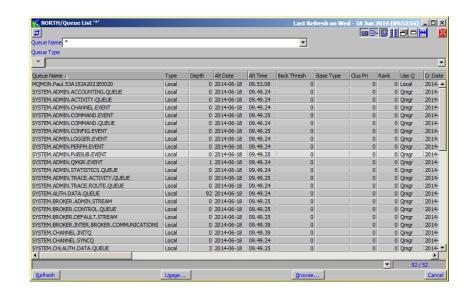
#### MQSCX

- Feature Overview
- New features
- Demo

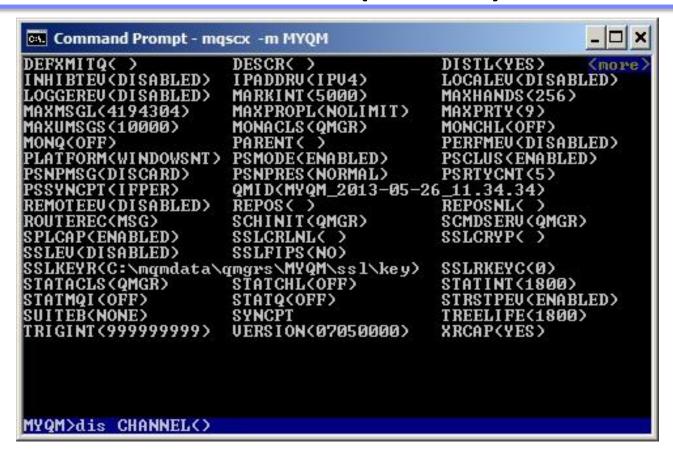
#### MO71

- Feature Overview
- New features
- Demo





# MQSCX (9.0.2)



#### Overview



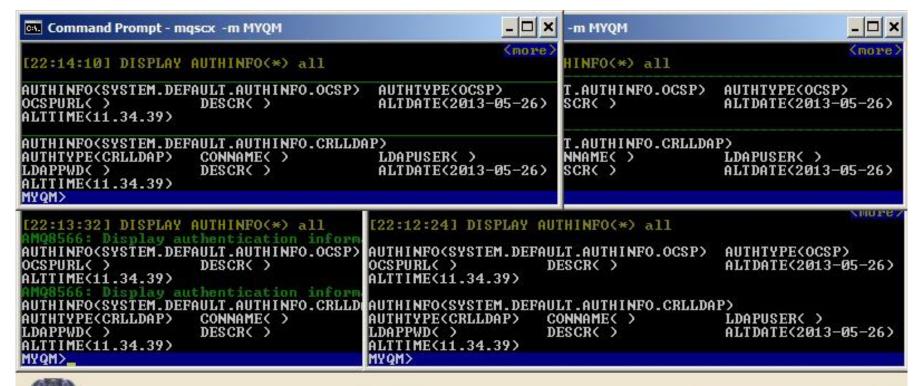
- Like RUNMQSC, but with so much more
  - Can be used interactively or in background
  - Fills many of the 'holes' in RUNMQSC
  - Runs as local or client bindings
  - More granular security
  - Less frustrating to use
  - More efficient and timely
  - Much Greater capabilities

```
Command Prompt - mqscx -m MYQM
    BTEU(DISABLED)
                    IPADDRU(IPU4)
                                          LOCALEU(DISABLED)
                     MAXPROPL(NOLIMIT)
                     MONACLS (QMGR)
   JMSGS(10000)
                     PARENT (
                                           PERFMEU(DISABLED)
                    PSMODE(ENABLED)
                     PSNPRES(NORMAL)
                     QMID(MYQM_2013-05-26_11.34.34)
                                          REPOSNL( )
                                          SCMDSERU (QMGR)
SLKEYR(C:\mgmdata\gmgrs\MYQM\ss1\key)
TATMQI(OFF)
                                          STRSTPEU(ENABLED)
                     SYNCPT
                    UERSION(07050000)
MYQM>dis CHANNEL()
```

#### **Separator Lines**



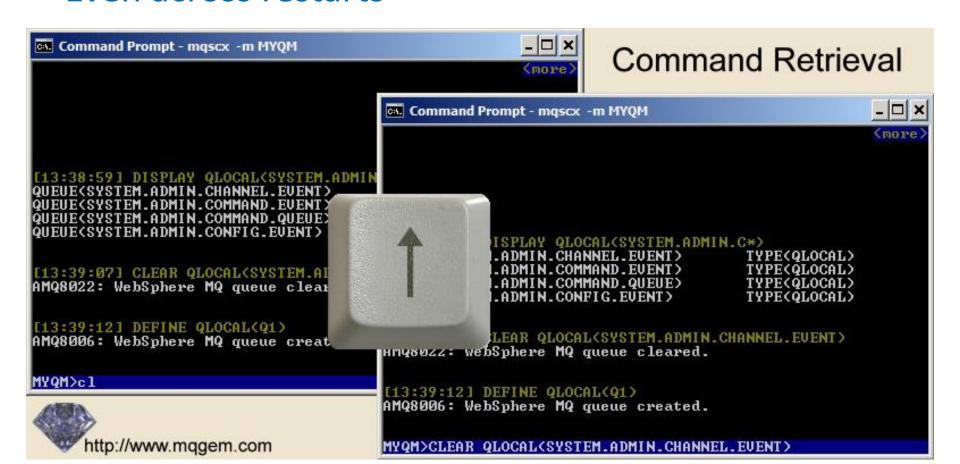
- Helps see the wood from the trees
- Can use significantly less screen real estate



#### **Command Retrieval**



- What is more annoying than no command retrieval?
- MQSCX allows generic and targeted retrieval.
- Even across restarts



### **Command Auto-Completion**



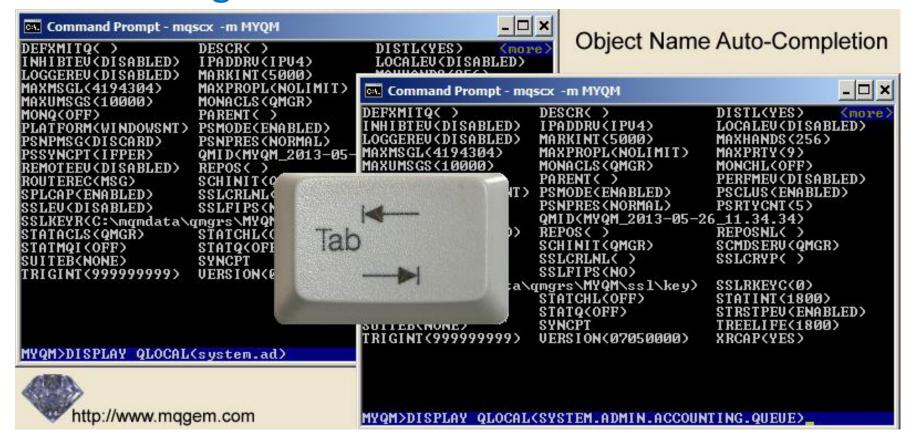
- Pressing 'Tab' will tab through available values
- Forwards and backwards supported
- Tab through commands, attributes or values
- Tab through like-named items



### Object name completion



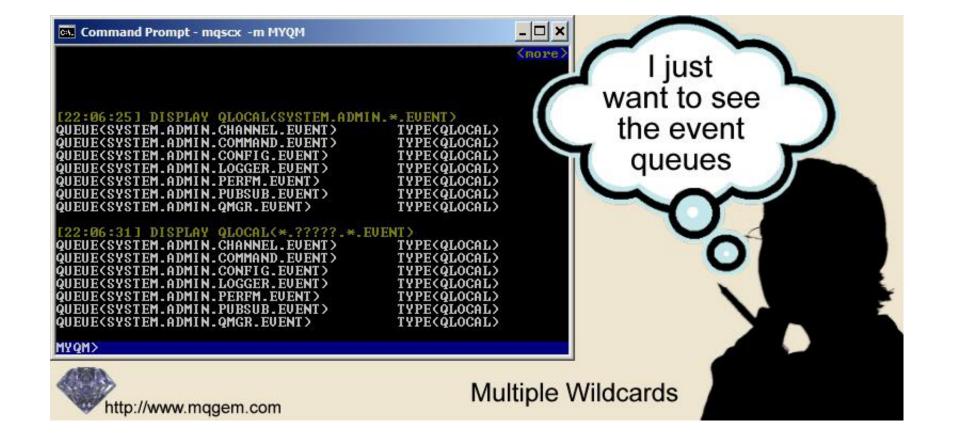
- Pressing 'Tab' will tab through available values
- Forwards and backwards supported
- Tab through like-named items



### Multiple Wildcards



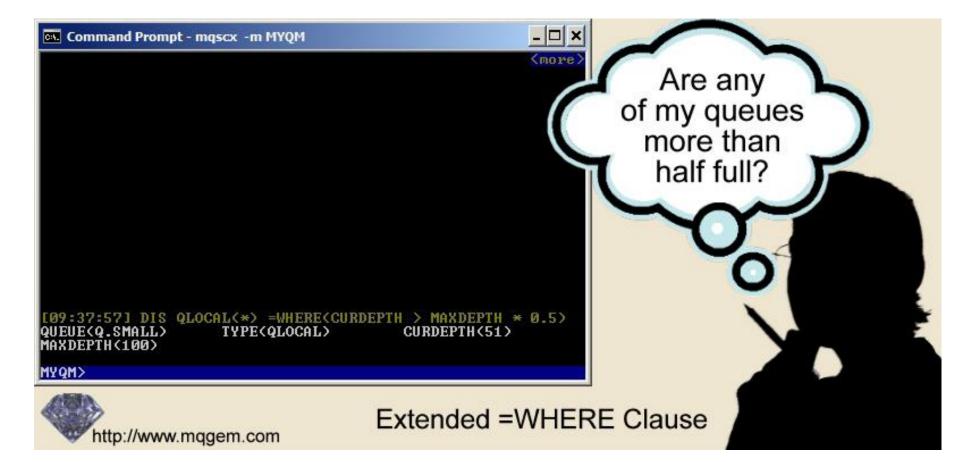
- Wildcards can be specified anywhere
  - Wildcards '\*' and '?' supported



#### Extended WHERE clause



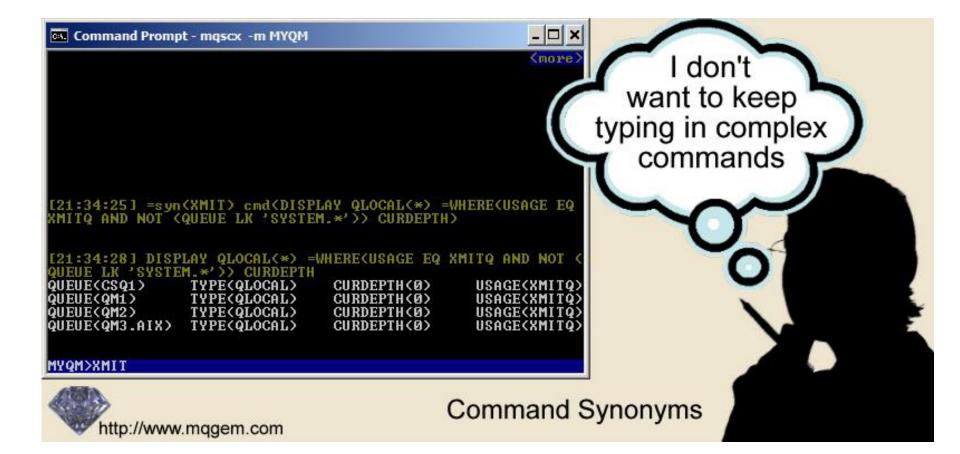
- Much more flexibility than standard WHERE
- Multiple expressions supported
- Greater list of operators
- Numeric expressions supported



#### Synonyms



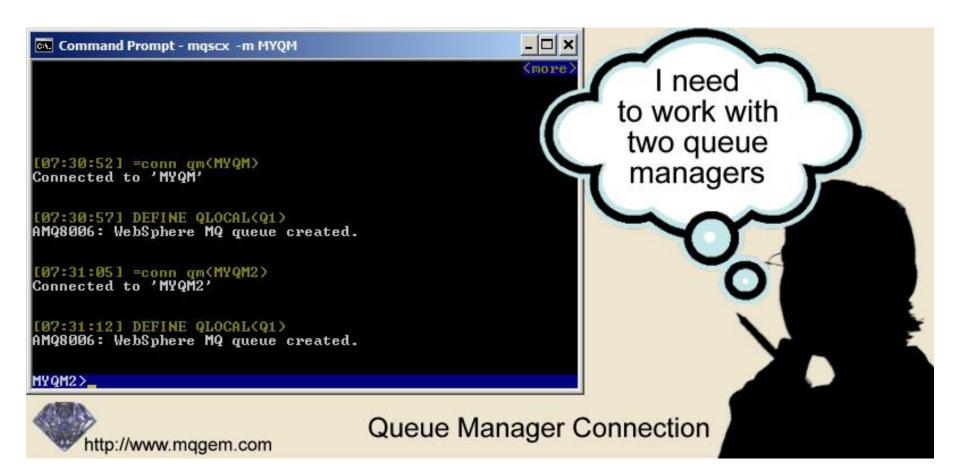
- Saves laborious typing
- Reduces the chances of making mistakes
- Can go further even define a single keystroke as a command



### Connecting and Disconnecting



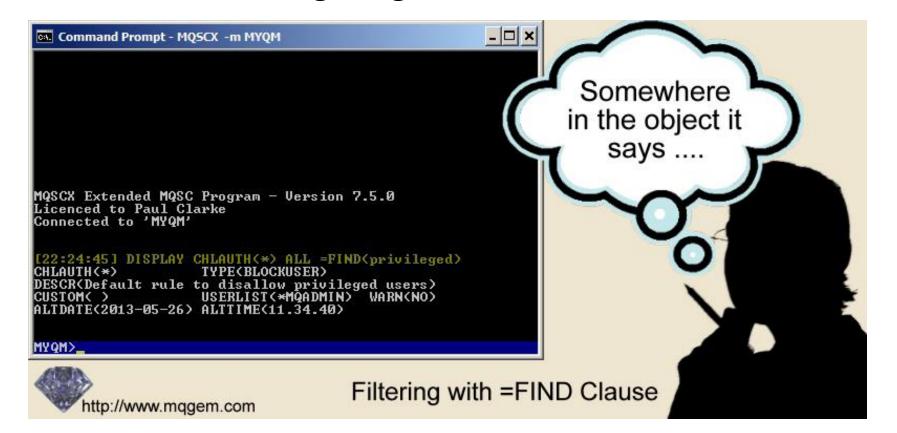
- Allows connecting and disconnecting
  - All within same script
  - Connect as client or local bindings
  - Prompt changes depending on connected Queue Manager



#### FIND clause



- Simpler way of just filtering by text
  - Text matched against any of the response
  - Much easier to get right than WHERE clause



#### Plus much more....



- Colour configuration
- Multiple commands on a single line
- Display totals
- Full prompt configuration
- Search through responses
- IMPORT other command files
  - Which can also IMPORT command files

•

### **CCDT Editing**



- MQSCX can be run in 'CCDT' mode
  - CLNTCONN channels can be defined, changed, deleted
  - CCDT can be created of any MQ version
  - Two CCDT files can be merged (by channel name)

```
CCDT commands directed to file 'C:\mqmdata\AMQCLCHL.TAB'

MQSCX Extended MQSC Program - Version 7.5.1

Licenced to Paul Clarke

Licence Location: Home
[11:00:191 dis chl(*) conname = where(not(channel == "SYS*"))

CHANNEL(PRED) CHLTYPE(CLNTCONN) CONNAME(localhost(1415))

CHANNEL(NTPGC1.SRU) CHLTYPE(CLNTCONN) CONNAME(LOCALHOST)

CHANNEL(X) CHLTYPE(CLNTCONN) CONNAME(198.162.45.67)

Total display responses - Received:4 Matched:3
```

## Control Language



#### MQSC is a very static language

- Very difficult to parse
- No control constructs
- Just a list of commands to execute
- Very hard to even tell whether a command worked!

#### MQSCX extends the language

- Easy to process responses
- Adds loops, if statements, list processing, goto
- User variables
- Functions

### Control Language



#### Extends MQSC uses enormously

- Augment or modify command responses
- Display consolidated result of multiple commands
- Perform health checking
- Perform monitoring
- Write report file of Queue Manager status
- Check objects against naming conventions
- Bulk updates
  - For example, change all objects with a particular value

## 'Top' Script



```
* Get the local queues and their depth and store in array
* First delete any previous results
delvar(@val)
* Now, get a list of queues with messages
@i = 0;
foreach(DIS QL(*) CURDEPTH WHERE(CURDEPTH GT 0))
  @i = @i + 1
  @val[1,@i] = QUEUE
  @val[2,@i] = CURDEPTH
endfor
* Sort the array in descending order by second column
sortd(@val,2)
* Print out a screen full unless there are even less returned
if (height)
                 @limit = height-3; else @limit = 20; endif
if (@limit > @i) @limit = @i; endif
* Ok, print out the top few
@i = 0;
print :48:"Queue Name","Depth"
print :48:"======","====="
while(@i < @limit)</pre>
  @i = @i + 1
  print :48:@val[1,@i],:5:@val[2,@i]
endwhile
```

#### ccdver function



```
* Determine the minimum and maximum version in a CCDT
func ccdtver()
if ( ccdtmode)
  0maxVer = 0
 foreach (DISPLAY CHANNEL(*) VERSION)
   if (VERSION < @minVer)</pre>
      @minVer = VERSION
    endif
   if (VERSION > @maxVer)
      @maxVer = VERSION
   endif
 endfor
 if ( numEach)
   print "Channel version in",_ccdt
   print "Minimum:",@minVer,"Maximum:",@maxVer
  else
   print "No channels found in", ccdt
  endif
else
 print "Run this script with MQSCX in CCDT mode"
endif
endfunc
```

#### 'Conns' Function



```
func conns()
 =echo resp(no)
 if ( connqmgr = '')
   print 'Please connect to your queue manager before issuing "conns"'
    return
  endif
 DISPLAY OMSTATUS CONNS
 print 'Total connections:',CONNS
  @totalconns = CONNS
 DISPLAY CONN(*) WHERE (CHANNEL EQ ' ')
  @chlconns = @totalconns - matches
 print ' Local :', matches
  @total = 0
  @svrcn = 0
  foreach (DISPLAY CHSTATUS (*) CURSHCNV CHLTYPE WHERE (STATUS EQ RUNNING))
   if (CHLTYPE = "SVRCONN")
      @svrcn = @svrcn + 1
      @total = @total + CURSHCNV
   endi f
  endfor
 print ' QMqr Chls :',@chlconns-@total
 print ' Client Chls:',@total
 print
 print 'Total Running Channel instances:', matches
 print '
                  Channels:', matches - @svrcn
           QMgr
 print '
          Client Channels: ',@svrcn
endfunc
```

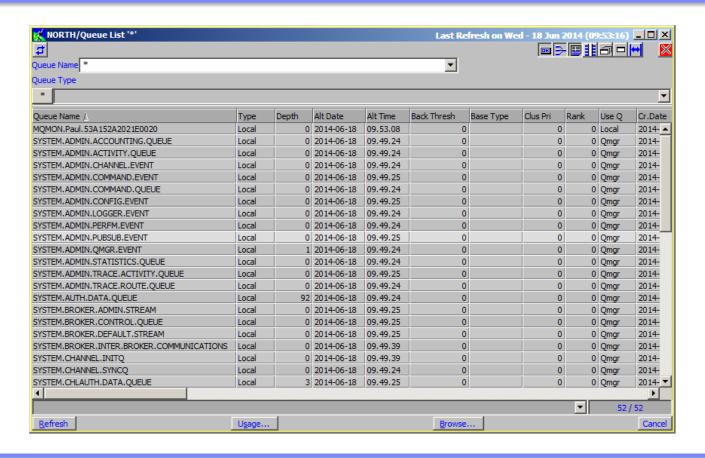
## MQSCX Conclusion



- MQSCX makes interactive and scripting easier
  - Command Line recall
  - Auto-complete of commands, attributes and object names
  - Command language
    - actually process the results of you commands
  - Multiple Queue Manager scripts
  - Improved user interface
    - Colours, sections, columns
  - Find results of commands
  - Improved WHERE clause
  - CCDT creation and editing (or any MQ version)

**Any Questions?** 

# MO71 (9.0.5)





#### MO71 Overview



#### Windows GUI

- Multiple window design
  - Show what you want, when and where ever you want it.
- Single EXE install, small (~3MB) and fast

#### Been around for 20 years

- Used by hundreds of customers all over the world
- Support any version up to MQ V9.0

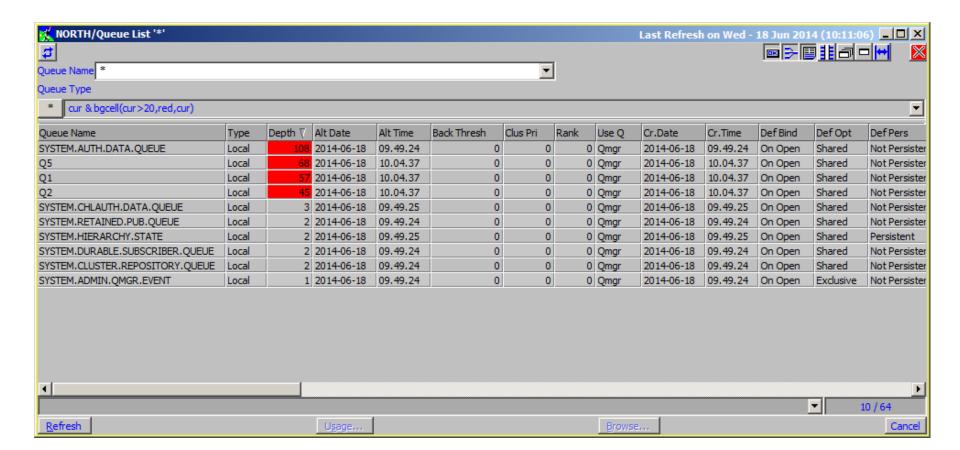
#### Very extensive list of features

- Not just configuration
- Monitoring and Administration

#### Filtering



- Very simple and yet very powerful filtering
  - Simple Boolean expression
  - Expression controls what and how fields are displayed



## Filter Manager



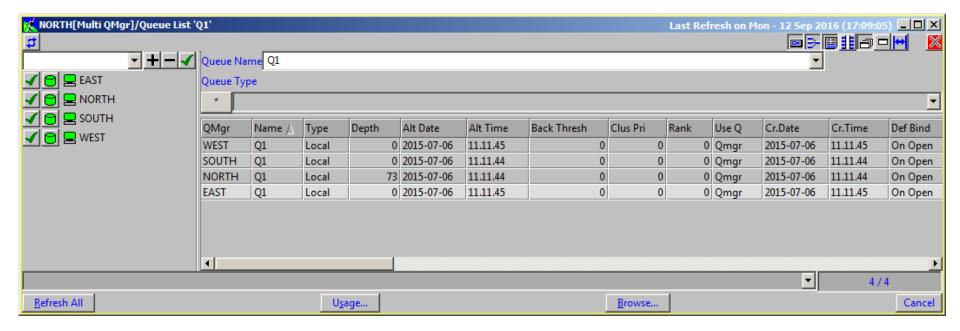
- Allows a name to be assigned to each filter
  - Handy if it's complicated
  - Or just to save typing
  - Each filter given a name and invoked with \$filtername
  - For example, writedepth below:

```
if (_first)
{
    @fd:= fopen("c:\\temp\\qs.csv","w");
}
fprintf(@fd,"%s,%s,%d\n",date$(_time),queue,curdepth);
if(_last)
{
    fclose(@fd);
}
1;
```

### Multi-Queue Manager Dialogs



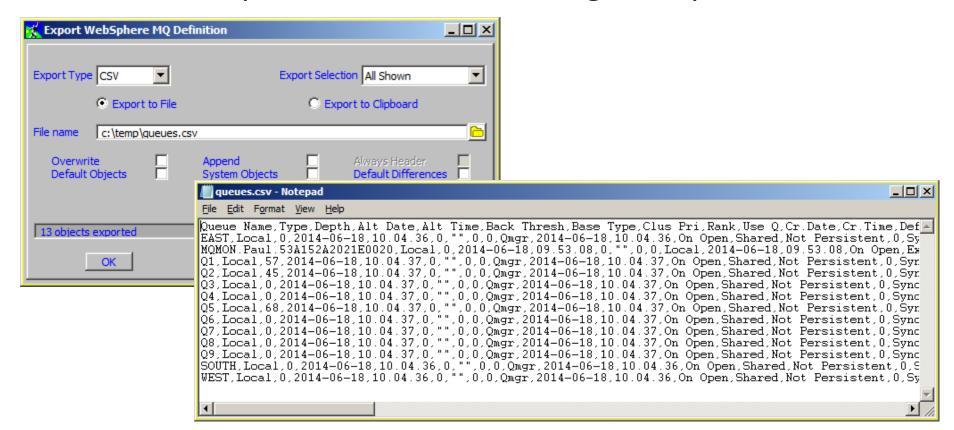
- Shows the results for multiple Queue Managers
  - In the same dialog
  - Subject to the same filters
  - Uses less screen real-estate



#### **Export Objects**



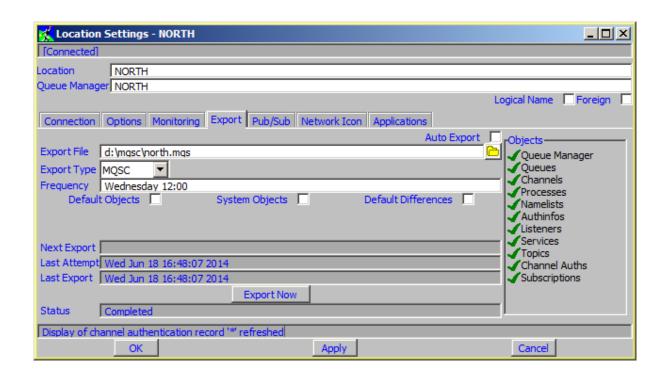
- All lists of MQ objects can be exported
  - To a file or the clipboard
  - In text, MQSC, CSV or XML formats
  - Various options to control what gets exported



# Exporting the whole Queue Manager



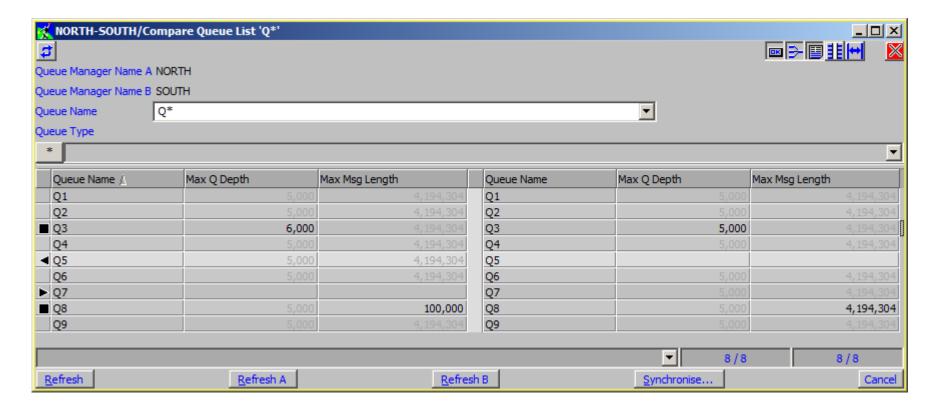
- Will save all the object definitions
  - Can be scheduled to run at any frequency
  - Again you can choose what gets exported
    - Such as default objects, system objects.



### **Comparing Definitions**



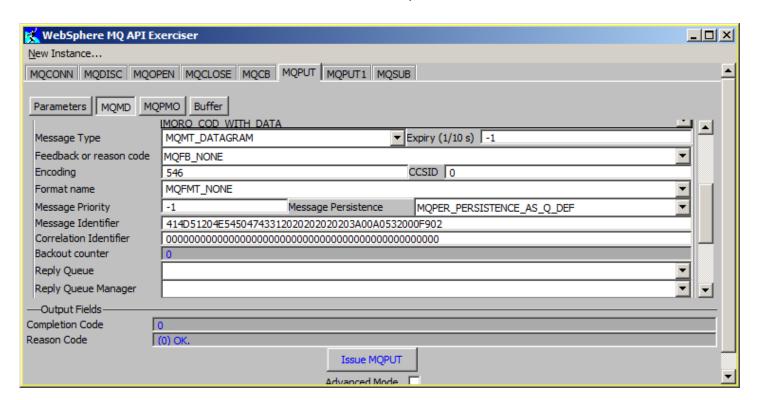
- Compare definitions on any two Queue Managers
  - Shows which objects are different
  - Shows which attributes are different
- Allows object definitions to be synchronised



#### **API Exerciser**



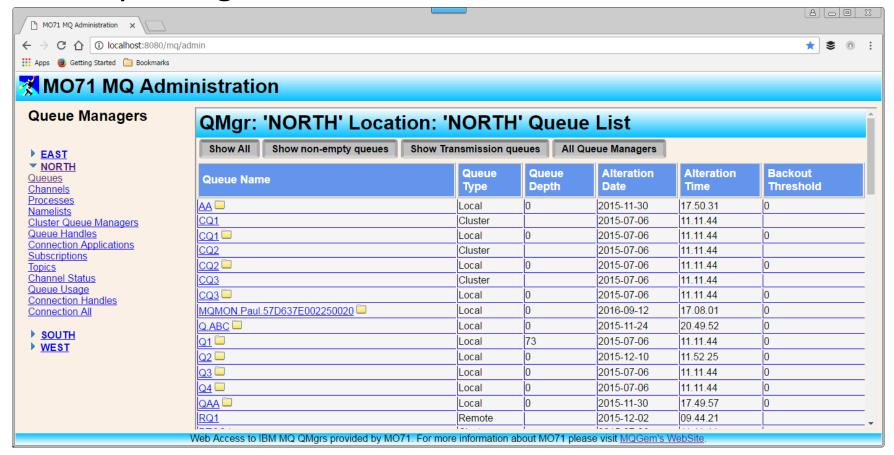
- Allows you to try out most of the MQI
  - Try out MQI combinations
  - See the effect or generate error conditions
  - Supports multiple local or client connections
  - Shows the whole of the MQI structures



#### **Browser Access**



- View your MQ status from anywhere
  - Read-only
  - Fully configurable



#### **Event Monitoring**

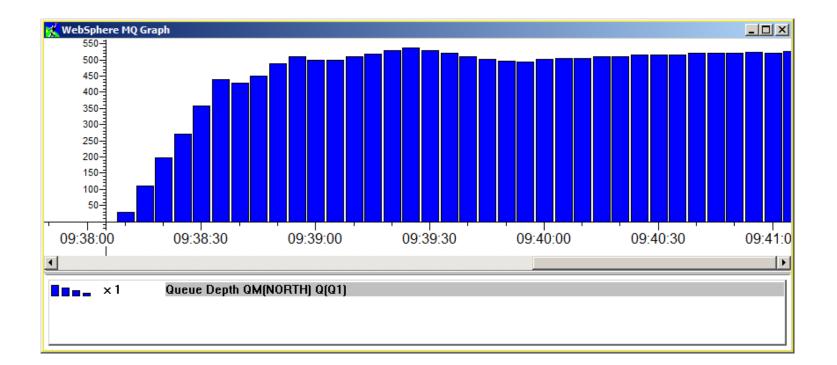


- Will 'listen' on any event queue for given event
  - Can filter by any event field
- Choice of event actions
  - Log to console
  - Log to a file
  - Send to another queue
  - Issue a command
    - Can be used to send email or pager etc

### Graphing



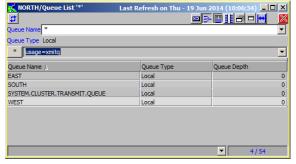
- Monitor virtual any numeric value
  - Set frequency, graph type etc
  - Graph multiple objects and object types
  - Export data-points to CSV

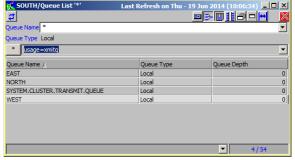


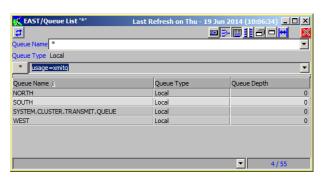
#### **Predefined Dialogs**



- Allow you to pre-set dialog...
  - Content
  - Even Multi-Queue Manager supported via qm() filter fn.
  - Filtering
  - Position and Size
  - Refresh rate
- Groups supported
  - Any number of dialogs can be started with single click



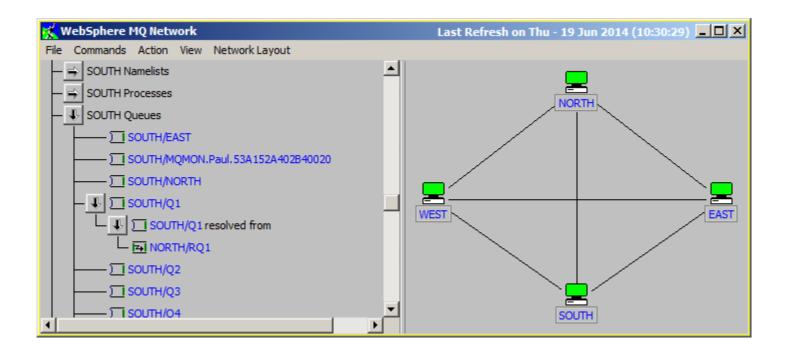




### **Network Display**



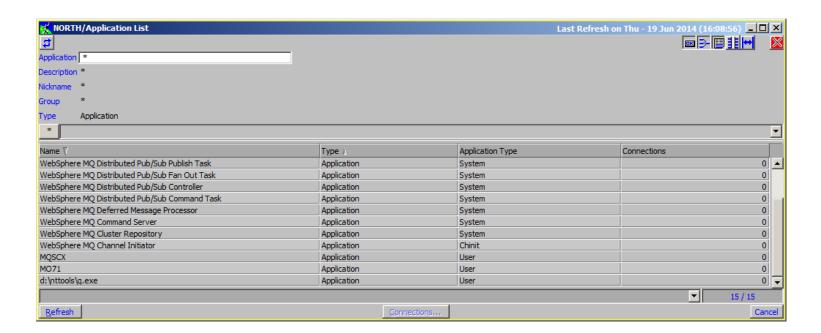
- Diagrammatic view of your network
  - Position Queue Managers with your own picture
- Verify view
  - Shows object resolution both to and from
  - Shows any health-check problems



## **Application List**



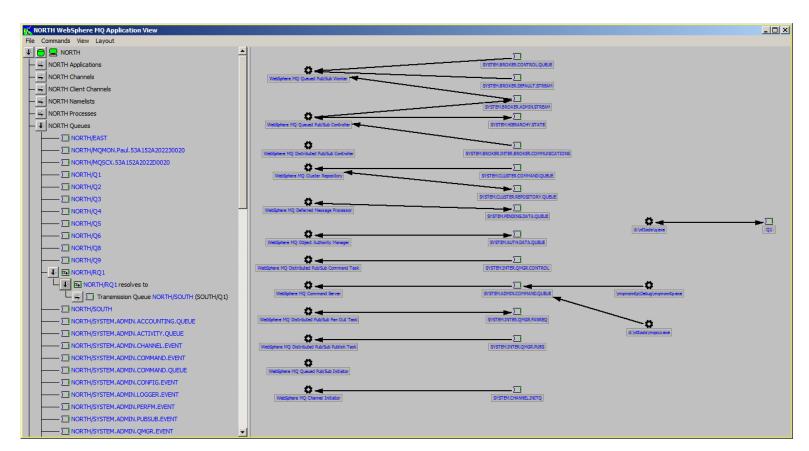
- List of applications
  - Generate dynamically from monitoring
  - Show status, inactive, running or error
  - Shows history of objects opened by the application
    - Including last used times



### **Application View**



- Shows status of applications in a diagram
  - Choose what applications to show...and where
  - Whether application is active and/or has queue open
  - User picture can be assigned to an application



## Thank-you

## Any Questions?

If you would like try either (or both) product(s) yourself then please just send an email to <a href="mailto:support@mqgem.com">support@mqgem.com</a> and a free, no-obligation one-month licence will be sent to you for your evaluation.



www.mqgem.com

support@mqgem.com