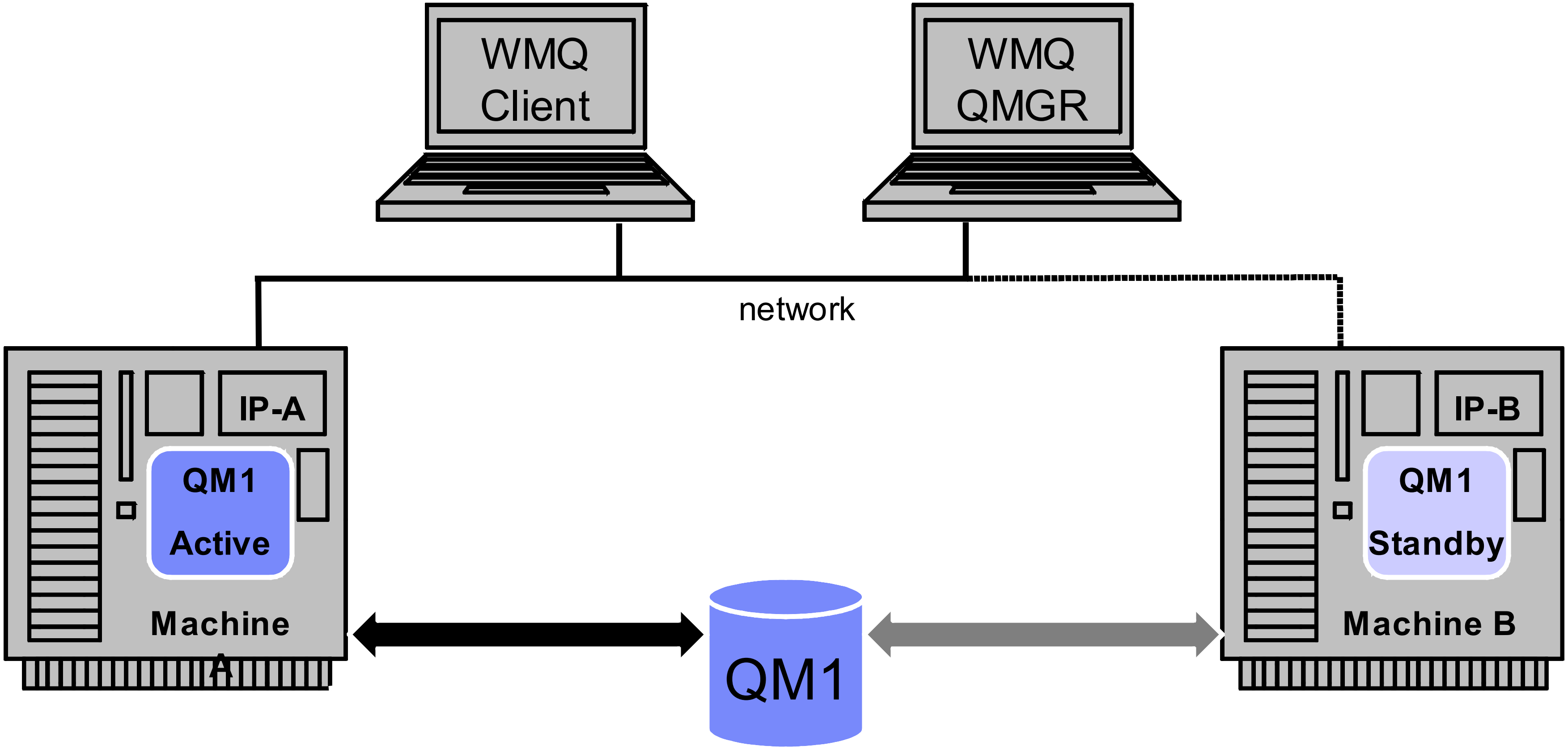
# Implementation Considerations for Multi-Instance Queue Managers in WebSphere MQ Cluster Environment

Yana Johnson – yanar@us.ibm.com
Advisory Software Engineer, IBM®

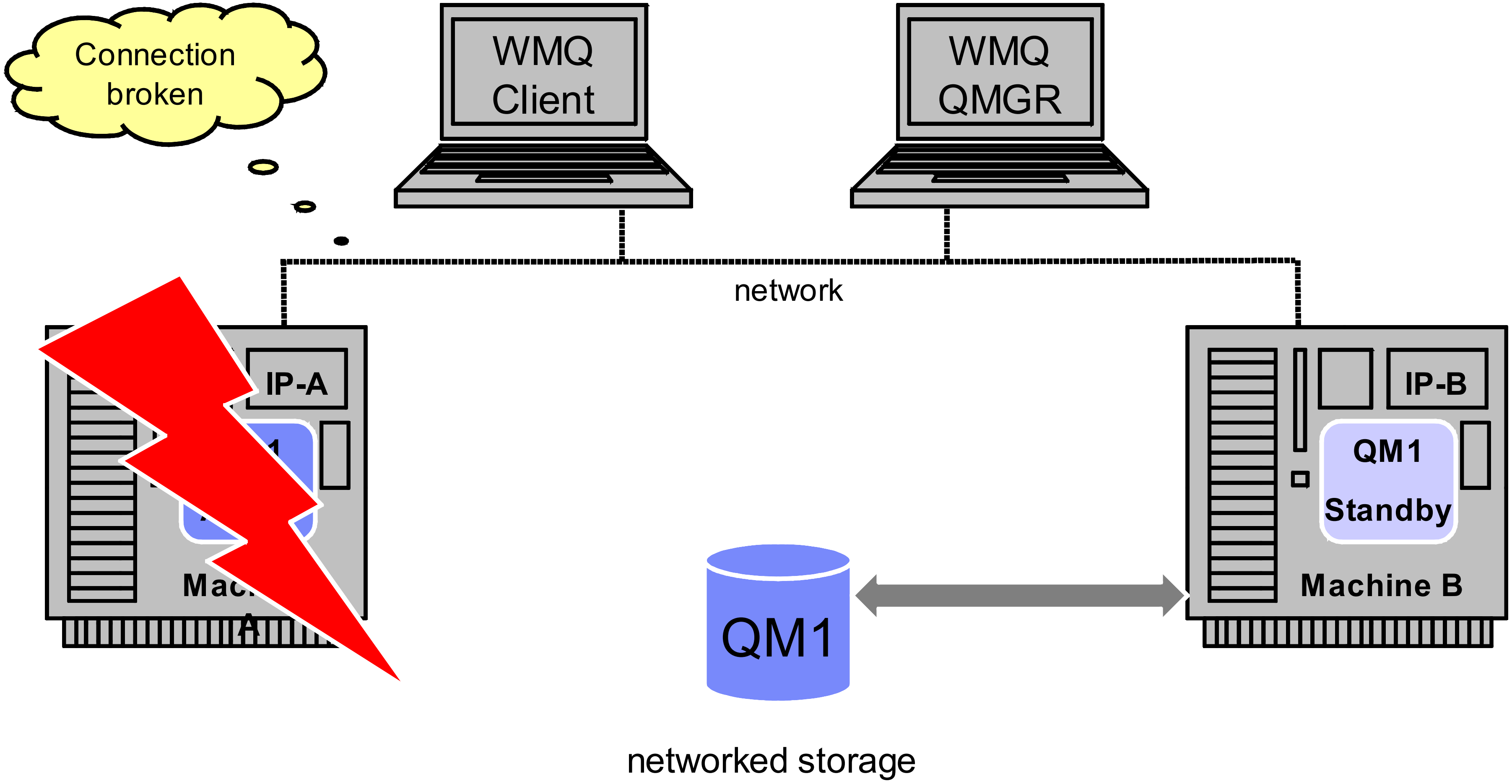
# Agenda

- What is a multi-instance queue manager?
- Implementing a multi-instance queue manager in a cluster
  - Limitations
  - Recommendations
  - Examples
- Summary
- Q & A

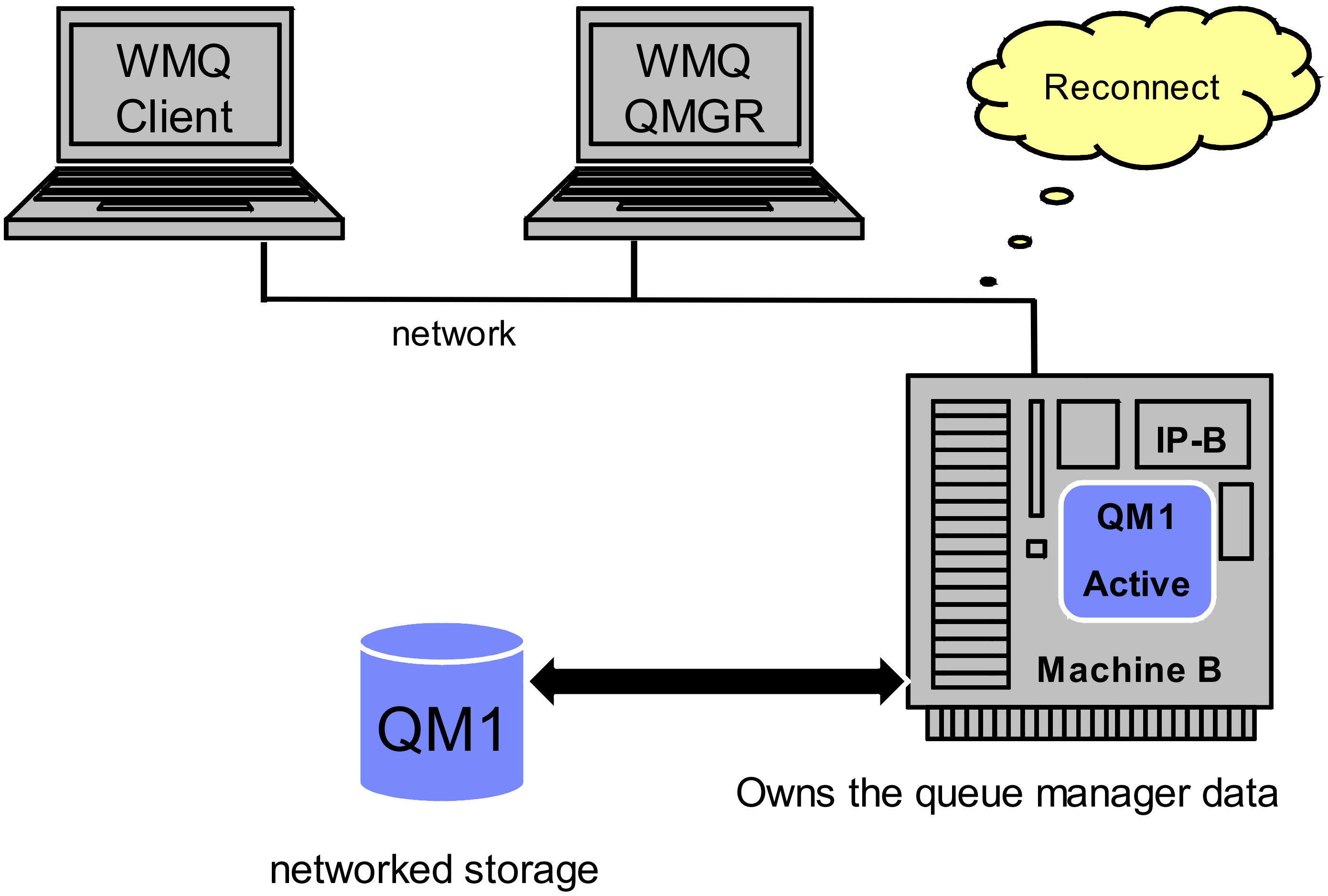
# Basic multi-instance operation



# Machine A system failure



# Reconnect – recovery complete



# Additional Information

- WSTE – An Introduction to WMQ Multi-Instance Queue Managers

  http://www-01.ibm.com/support/docview.wss?rs=233&uid=swg27017382

- How to convert queue manager to be multi-instance? (UNIX® only!)

# Conversion Steps

- This process should be performed by the root user. It is also best, even if it is just for the migration process, that the server mounting the NFS filesystem is provided root access of the filesystem.

- Assume that the qmgrs and logs exist in the default directory:
  - /var/mqm/qmgrs and /var/mqm/log

- Assume that the NFS share for multi-instance queue manager support is:
  - /MultiIn/qmdata and /MultiIn/logdata

- Assume that the queue manager we wish to move is called MY.QMGR

# Conversion Steps (continued)

- Copy the queue managers directories to the NFS filesystem.
  - cp -hpR /var/mqm/qmgrs/MY!QMGR /MultiIn/qmdata
  - cp -hpR /var/mqm/log/MY!QMGR /MultiIn/logdata
- Retrieve the queue manager information from the mqs.ini file
  - dspmqinf MY.QMGR

```
QueueManager:
         Name=MY.QMGR
         Directory=MY!QMGR
         Prefix=/var/mqm
```

- Remove the queue manager from the mqs.ini
  - rmvmqinf MY.QMGR

# Conversion Steps (continued)

- Add the queue manager back into the mqs.ini file
  - addmqinf -v Name=MY.QMGR -v Directory=MY!QMGR -v Prefix=/var/mqm -v DataPath=/MultiIn/qmdata/MY!QMGR
- Verify the changes to the mqs.ini
  - dspmqinf MY.QMGR

```
QueueManager:
          Name=MY.QMGR
          Directory=MY!QMGR
          Prefix=/var/mqm
          DataPath=/MultiIn/qmdata/MY!QMGR
```

# Conversion Steps (continued)

- Edit the qm.ini for the queue manager and update the LogPath entry in the Log: stanza.  This file will now be located in the /MultiIn/qmdata/MY!QMGR directory.

```
Log:
        LogPrimaryFiles=3
        LogSecondaryFiles=2
        LogFilePages=1024
        LogType=CIRCULAR
        LogBufferPages=0
        LogPath=/MultiIn/logdata/MY!QMGR/
        LogWriteIntegrity=TripleWrite
```

- Verify the startup and stopping of the queue manager using a normal startup.
  - strmqm MY.QMGR
  - endmqm -i MY.QMGR

# Conversion Steps-Cleanup

- Although not necessary you may perform the following two cleanupsteps to remove unnecessary files and directories.

- Remove the following files from the /MultiIn/qmdata/MY!QMGR directory.
  - @app
  - @qmpersist
  - AMQRSYNA.DAT
  - esem
  - isem
  - msem
  - qmgrlocl
  - shmem
  - spipe
  - zsocketEC
  - zsocketapp

# Conversion Steps–Cleanup (continued)

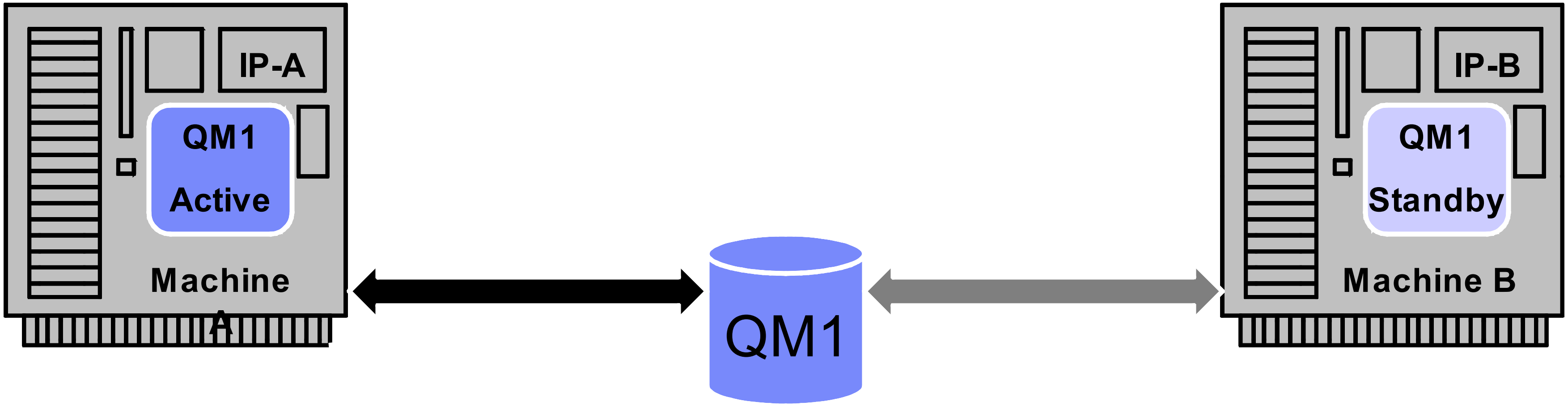
- Once you have verified queue manager startup/shutdown you can then cleanup files in the /var/mqm/qmgrs/MY!QMGR directory. All files and directories can be removed except the following:
    - @app
    - @ipcc
    - @qmgr
    - @qmpersist
- Remove the log files and directory
    - rm –r /var/mqm/log/MY!QMGR
- Conversion is now complete.

# Agenda

- ✓ What is a multi-instance queue manager?
- ➢ Implementing a multi-instance queue manager in a cluster
  - ➢ Limitations
  - Recommendations
  - Examples
- Summary
- Q & A

# Limitations

- System IP address is not carried over during a failover
- Will effect communication between 7.0.1 and older queue managers



# Limitations

- Connection names
  - In 7.0.1

```
define channel(TO.MACHINE_A)chltype(CLUSRCVR)

conname('MACHINE_A(port),MACHINE_B(port)')  cluster(MULTI)


define channel(TO.MACHINE_A) chltype(CLUSSDR)
conname('MACHINE_A(port),MACHINE_B(port)') cluster(MULTI)
```

  - Pre - 7.0.1

```
define channel(TO.MACHINE_A) chltype(CLUSRCVR)

conname('MACHINE_A(port)') cluster(MULTI)


define channel(TO.MACHINE_A) chltype(CLUSSDR)

conname('MACHINE_A(port)') cluster(MULTI)
```

# Limitations

- Multi instance queue manager is a full repository
  - Pre - 7.0.1 repositories will have to have a channel to both machines that Multi-Instance queue manager is using
  - Stuck messages on SYSTEM.CLUSTER.TRANSMIT.QUEUE

# Limitations

- z/OS
  - CONNAME attribute 48 character limit
    - Long hostnames may not fit
  - Workaround
    - Set up DNS servers to use hostname "myserver" instead of "myserver.location.company.com"
    - Use IP addresses

# Agenda

- ✓ What is a multi-instance queue manager?
- ➢ Implementing a multi-instance queue manager in a cluster
  - ✓ Limitations
  - ➢ Recommendations
  - ▸ Examples
- Summary
- Q & A

# Recommendations

- Use listener service so that listener(runmqlsr) will start automatically as part of queue manager startup

  DEFINE LISTENER(NAME) TRPTYPE(TCP) CONTROL (QMGR) PORT(PORT NUMBER)

- Define only two full repositories

- Do NOT use multi-instance queue managers as full repositories

# Recommendations

- If a multi-instance queue manager(s) is part of a cluster, upgrade all queue managers within that cluster to 7.0.1 (ideal scenario!)
- If upgrade to 7.0.1 is not possible, use blank CONNAME on a 7.0.1 multi-instance queue manager
  Use default port only!

# Agenda

- ✓ What is a multi-instance queue manager?
- ➢ Implementing a multi-instance queue manager in a cluster
  - ✓ Limitations
  - ✓ Recommendations
  - ➢ Examples
- Summary
- Q & A

# Examples

- Cluster Queue managers:

Full repositories: QM1, QM2
Partial repositories: QM3
Multi-Instance partial repository: MIQMGR

- Case #1: All clustered queue managers are 7.0.1

```
DEFINE CHANNEL(TO.MIQMGR) CHLTYPE(CLUSRCVR)
CONNAME('MACHINE_A(PORT),MACHINE_B(PORT)') CLUSTER(MULTI)

DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSSDR) CONNAME('MACHINE_C(PORT)')  CLUSTER(MULTI)
```

- Case #2: Some queue managers are 7.0.1

```
DEFINE CHANNEL(TO.MIQMGR) CHLTYPE(CLUSRCVR) CONNAME(' ') CLUSTER(MULTI)

DEFINE CHANNEL(TO.QM1) CHLTYPE(CLUSSDR) CONNAME('MACHINE_C(PORT)')  CLUSTER(MULTI)
```

# Agenda

- ✓ What is a multi-instance queue manager?
- ✓ Implementing a multi-instance queue manager in a cluster
  - ✓ Limitations
  - ✓ Recommendations
  - ✓ Examples
- ➢ Summary
- Q & A

# Summary

- Brief introduction to Multi Instance queue managers
- Limitations and recommendations when implementing a Multi Instance queue manager within an WMQ cluster
- Configuration examples

# Additional WebSphere Product Resources

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/
- Learn about other upcoming webcasts, conferences and events: http://www.ibm.com/software/websphere/events_1.html
- Join the Global WebSphere User Group Community: http://www.websphere.org
- Access key product show-me demos and tutorials by visiting IBM Education Assistant: http://www.ibm.com/software/info/education/assistant
- View a Flash replay with step-by-step instructions for using the Electronic Service Request (ESR) tool for submitting problems electronically: http://www.ibm.com/software/websphere/support/d2w.html
- Sign up to receive weekly technical My support emails: http://www.ibm.com/software/support/einfo.html

# Agenda

- ✓What is a multi-instance queue manager?
- ✓Implementing a multi-instance queue manager in a cluster
  - ✓Limitations
  - ✓Recommendations
  - ✓Examples
- ✓Summary
- ➢Q & A

# We Want to Hear From You!

**Tell us about what you want to learn**

Suggestions for future topics
Improvements and comments about our webcasts
We want to hear everything you have to say!

**Please send your suggestions and comments to:**
**wsehelp@us.ibm.com**

# Questions and Answers