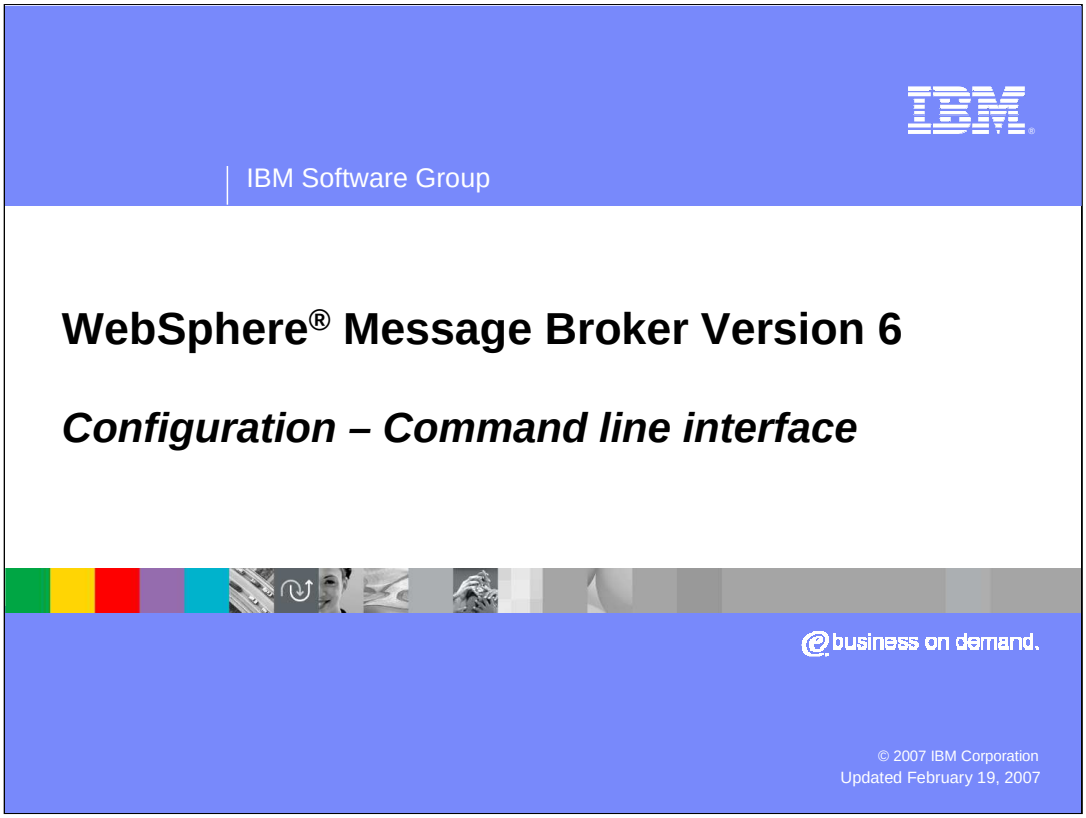
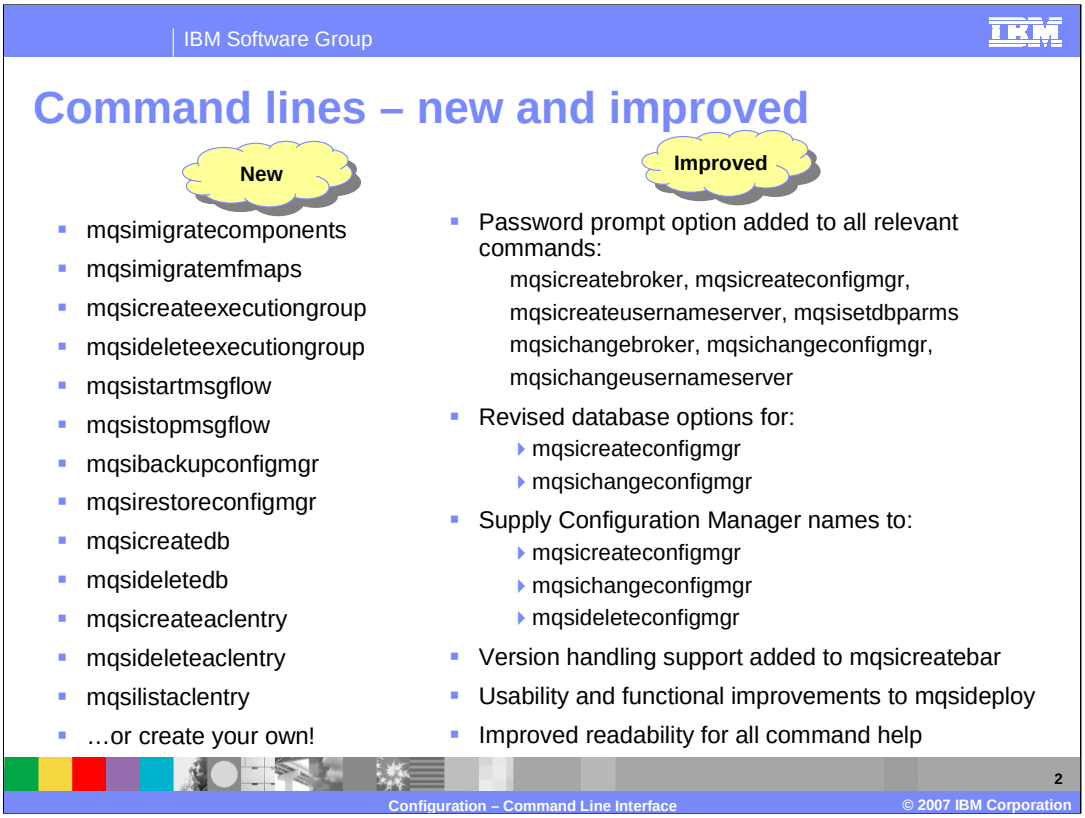


This presentation discusses the new and improved Command Line Interface in WebSphere Message Broker Version 6.



This slide shows some of the new and improved commands provided in WebSphere Message Broker Version 6. For a complete list of line commands search topic an07060 in the Information Center.

As a result of the ability to create multiple Configuration Managers, the correct profile must be set before using the line commands. The profile is set with the “mqsiprofile” command. On Windows®, this is provided by running these commands in the command prompt windows opened when you open the Broker Command Console.

Password prompt option

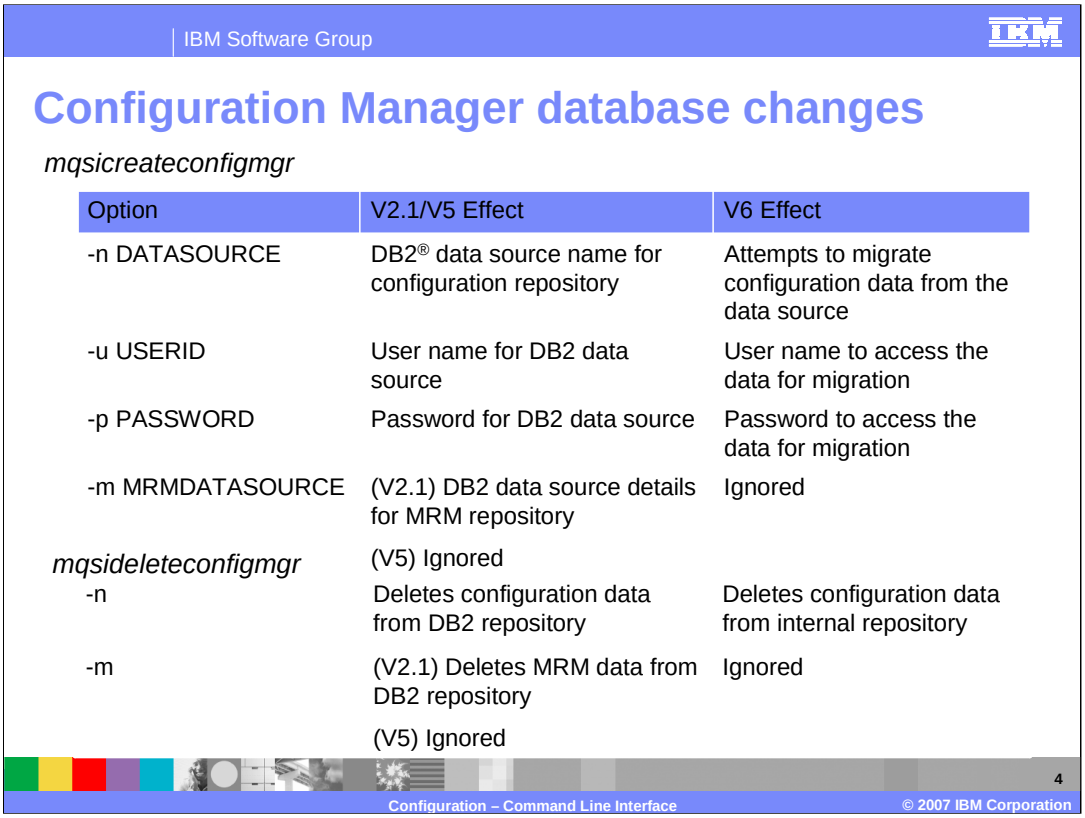
- All password parameters will now prompt the user for a password if one is not supplied
- Password is not revealed on the screen

```
> mqsicreatebroker BROKER -i userid -a -q
QMGR -n DB

Enter password for userid
Retype password for userid
WebSphere MQ queue manager created.
...
BIP8071I: Successful command completion.
>
```


One of the security enhancements that was made for Version 6 is the password prompt option. If the password is not provided on the command itself, the password will be prompted as part of the command execution. The password is not shown on the screen as it is entered.

The example on this slide shows the creation of a broker. Since the password is not included on the command, it must be entered when prompted.



A key change with WebSphere Message Broker Version 6 is the ability to create multiple instances of the Configuration Manager. Also, the Configuration Manager does not now store its information in an external database. This has been replaced by an internal database, contained entirely within the Configuration Manager.

This changed the behavior of commands that are directly related to the Configuration Manager DB2 database. This slide illustrates this with the “mqsicreateconfigmgr” command. The options have different meanings when applied to Version 6, compared with earlier versions.

In Version 5, the –n parameter was used to identify the Configuration Manager DB2 database. The command then used this to create a set of database tables in the database, which were used by the Configuration Manager. In Version 6, since there is no need to create an external database, the –n parameter is no longer required. In Version 6, this parameter is used to reference the Version 5 or Version 2.1 database, so that the Configuration Manager can be migrated up to the Version 6 level.

The –u and –p parameters are used in a similar manner. In Version 2.1 and 5, they are used to create the tables in the database. In Version 6, as with the –n parameter, they are used to access the old databases, for the purposes of migration to Version 6.

The –m parameter is used only in Version 2.1, where message set definitions were stored in the DB2 MRM database. In Version 6, as with Version 5, these definitions are stored in the Toolkit workspace, or in an external code management repository.

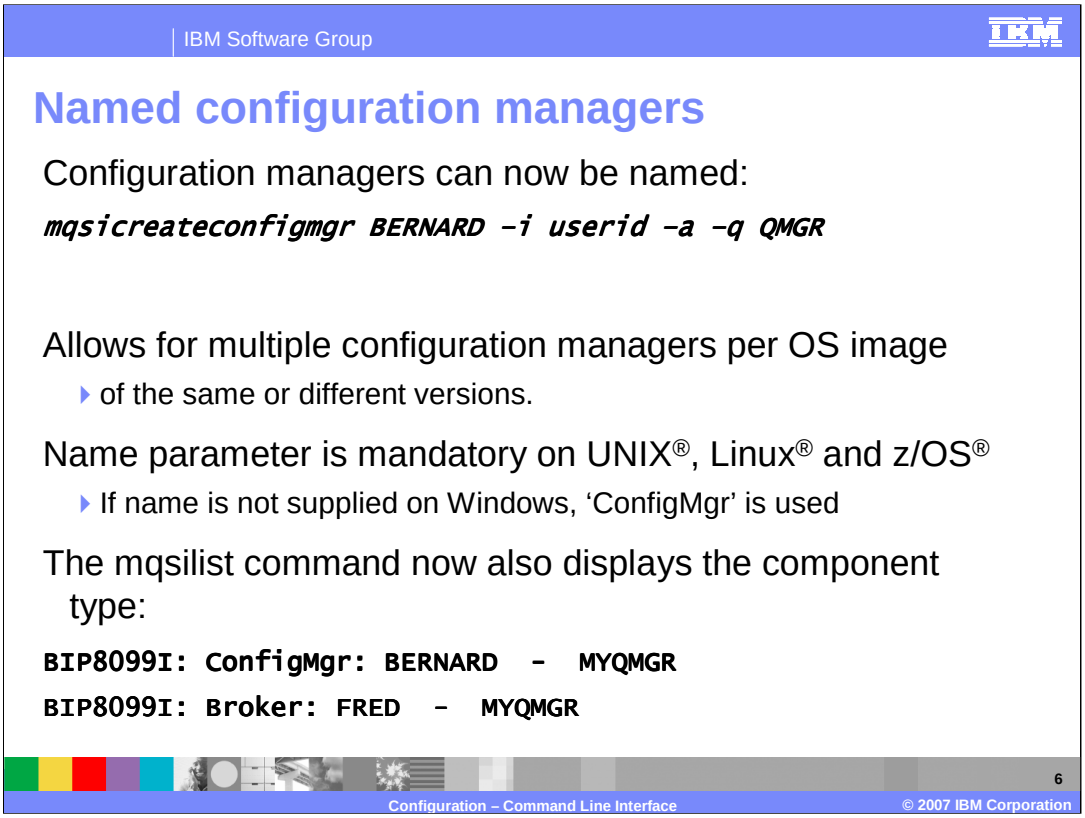
The –n parameter is used to delete data from the Configuration Manager database. In the case of Versions 2.1 and 5, data is deleted from DB2. In the case of Version 6, data is deleted from the internal database.

The –m parameter is used to delete MRM definition data; this is not required in Version 6.

When to use database options on *mqsicreateconfigmgr*

- The mqsicreateconfigmgr database options (-n -u -p) are now only required if you want to migrate a V5 DB2 repository into a new Configuration Manager
- This is because, when migrating a V5 Configuration Manager using mqsimigratecomponents, the DB2 data source is picked up by the V6 Configuration Manager and the data is migrated to the internal repository automatically
- If the -n flag is specified but Configuration Manager information at the supplied data source cannot be found, a warning is reported but a clean Configuration Manager repository is created
  - This means that any existing user-defined scripts to create the Configuration Manager can be used unchanged

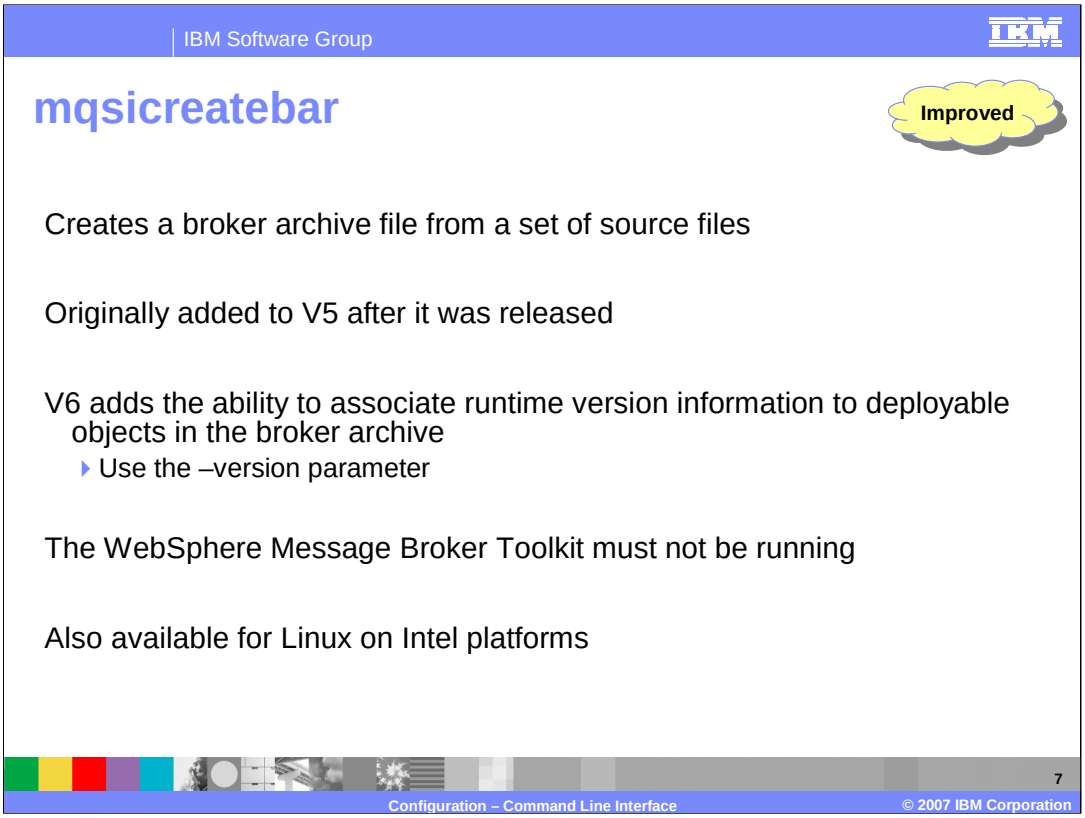
When performing a migration from Version 5, the “mqsimigratecomponents” is used to migrate user applications. This command will migrate Configuration Managers, brokers, and User Name Servers.



In V6 it is possible to create several Configuration Managers on a single system. This system may be Windows, as in earlier versions, but it is also possible to create the Configuration Manager on other platforms. The Configuration Manager may be located on the same system as the Broker runtime.

Since it is possible to have multiple instances of the Configuration Manager, the Configuration Manager must now be named, in order to be identified uniquely. When creating a Configuration Manager, the first parameter should be the name of the Configuration Manager. On the platforms which are supporting the Configuration Manager for the first time, namely UNIX, Linux and z/OS, this parameter is mandatory. On Windows, to maintain backward compatibility, it is optional, although of course if multiple Configuration Manager are required, they will need to be named uniquely.

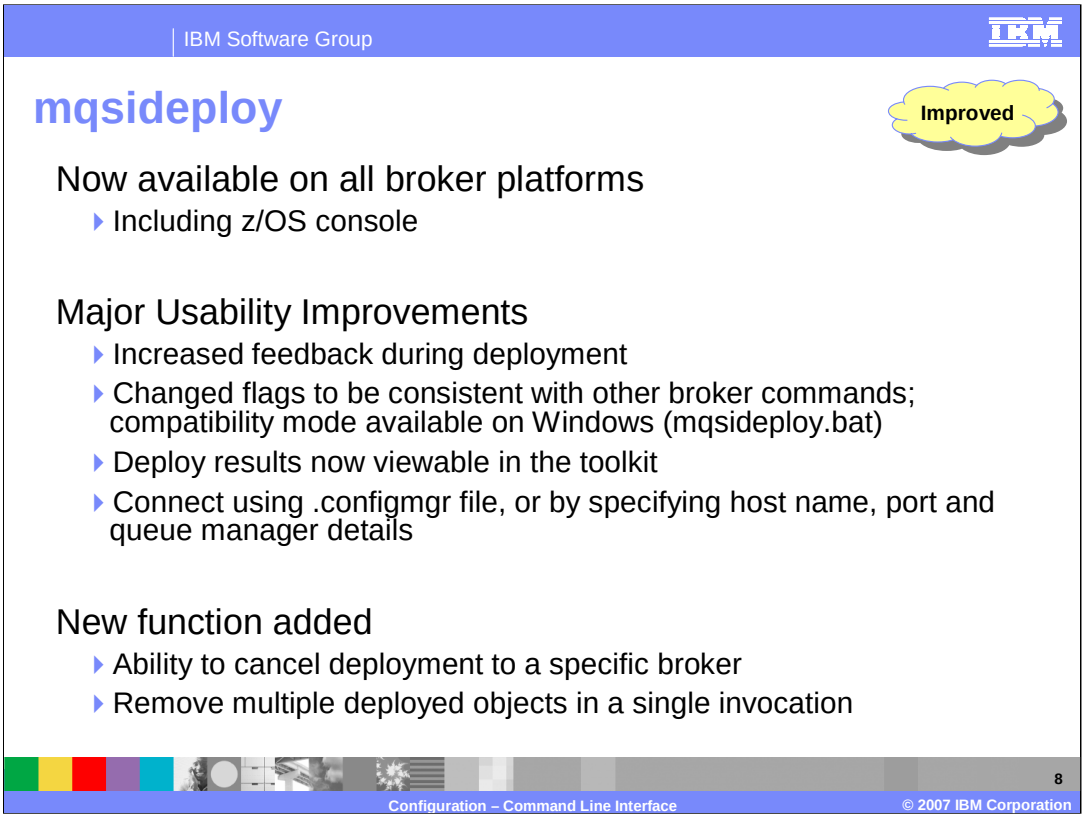
The “mqsilist” command has been updated to reflect this change, and now displays the type of resource and its name.



When creating a bar file for deployment, it may be necessary to do this without using the Broker Toolkit. The “mqsicreatebar” command enables the creation of a bar file in this way.

In Version 6, this command has been updated to associate a version number with the components of the bar file. The version handling capabilities of Message Broker are more fully described in another presentation.

Since this command uses some of the Eclipse facilities, the Broker Toolkit must not be running when this command is used. This and other WebSphere Message Broker Toolkit commands are found in the eclipse folder of your installation directory.



mqsideploy allows you to deploy a bar file into a production system without using the Broker Toolkit. This function has been extended to include the z/OS platform.

The usability of this command has been enhanced, and more information is provided on completion of the command. The next slide shows an example of the output from this command. When the command has been used to deploy a bar file to the broker runtime, the results of this deployment are shown in the Administration perspective of the Toolkit.

To use this command, the platform where the command is issued needs to have a connection to the appropriate Configuration Manager. The details of this connection are normally generated by the Toolkit, when the connection is first established. This activity creates a file with a suffix of ".configmgr". This file can be exported from the Toolkit and used by the "mqsideploy" command to establish its own connection to the Configuration Manager. Alternatively, when executing the command, it is possible to directly specify the connection details for the Configuration Manager.

The mqsideploy command also allows you to cancel a deployment request, and to remove all deployed objects in a single operation. For example, the command can be used to remove all deployed message flows in a single execution group.

Command line deployment - example

```
> mqsideploy -n c:\my.configmgr -l
BIP1044I: Connecting to the Configuration Manager's queue manager...
BIP1045I: Connecting to the Configuration Manager...
BIP1055I: Deploying pub/sub neighbors (topology) configuration...
BIP1098E: At least one failure response has been received.
BIP1058I: The Configuration manager supplied the following message(s):
              (...)
BIP1092I: Broker BR1 successfully processed the deployment request.
BIP1092I: Broker BR2 successfully processed the deployment request.
BIP1093E: Broker BR3 reported the following error(s):
              (...)
```


Here is an example of the “mqsideploy” command, showing the information that is returned on completion of the command.

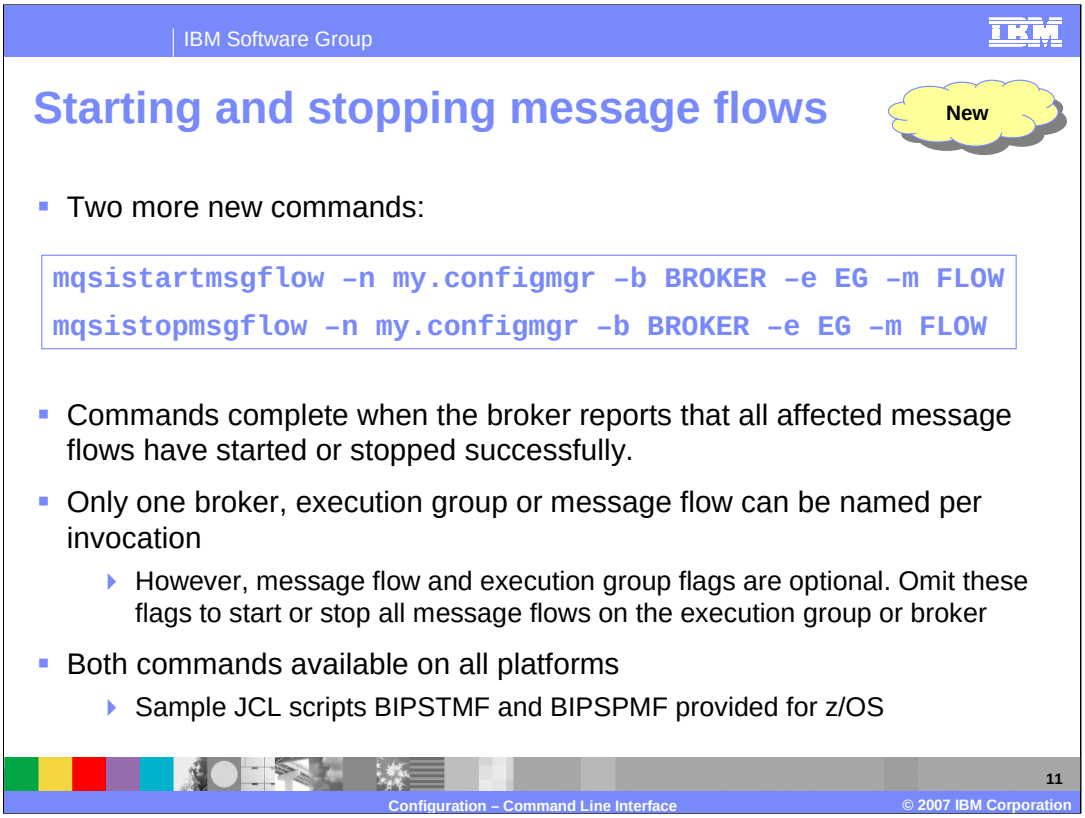
In this case, the command connected to the Configuration Manager by using the connection information stored in “my.configmgr”.

WebSphere Message Broker Version 6 introduces the facility to create and delete execution groups without using the Broker Toolkit as shown here.

When the execution group is defined, a record is made of this in the Configuration Manager. However, the execution group will not be fully defined until the first message flow is deployed to it. Hence, the execution group will not be visible in the Administration perspective of the Toolkit until a flow has been deployed.

When the execution group is deleted, any objects that have been deployed to that execution group will be lost. However, the source of those objects (message flows, sets, or the deployment bar file) will not be deleted.

As with the previous slide, these commands need to connect to the Configuration Manager. These examples use information stored in the file “my.configmgr”.



Message flows can be started and stopped using the two new commands shown on this slide. Again, a connection is required to connect to the Configuration Manager. These commands are then used to start or stop an individual message flow.

If the –m flag is omitted, then the command will be invoked for all message flows in the specific execution group.

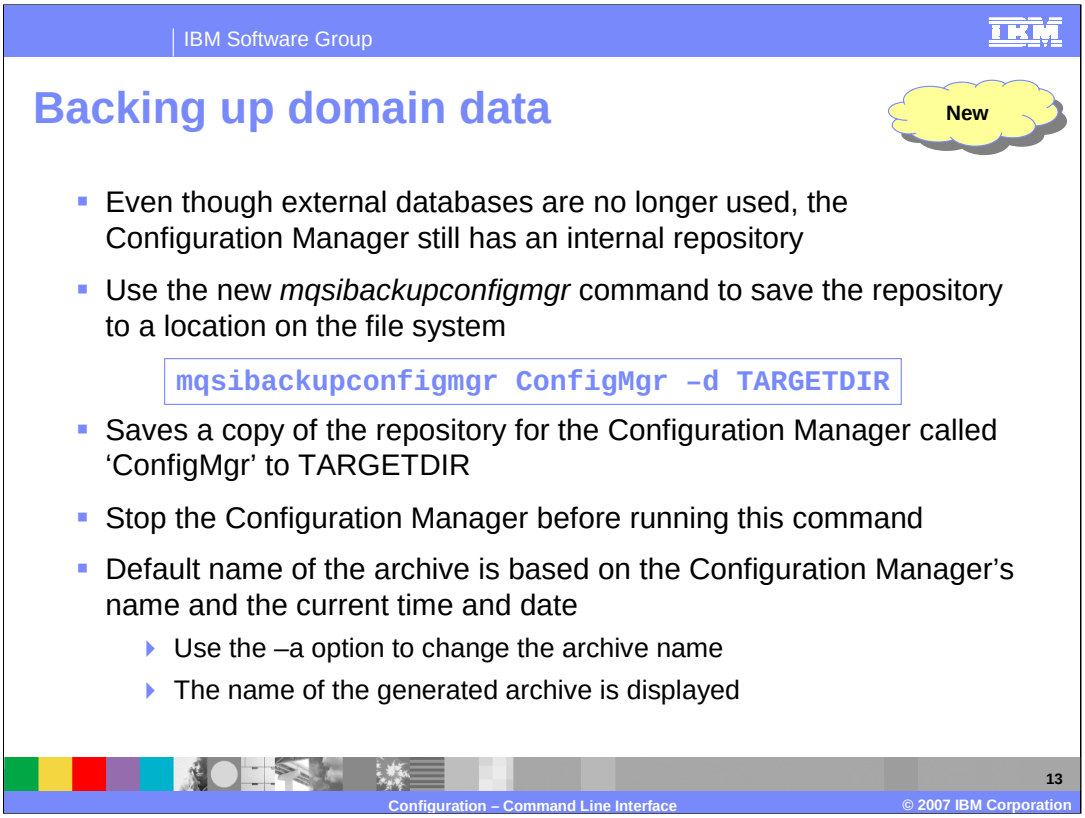
If the –e flag is omitted, then the command will be invoked for all message flows in the entire broker.

Stopping message flows - example

```
> mqsistopmsgflow -n c:\my.configmgr -b FRED
BIP1044I: Connecting to the Configuration Manager's queue manager...
BIP1045I: Connecting to the Configuration Manager...
BIP1028I: Submitting request to the Configuration Manager to stop
'FRED'...
BIP1029I: Waiting up to 60 second(s) for broker FRED to update its
configuration...
BIP1031I: Message flow 'FRED/default/flow1' is reported as stopped.
BIP1031I: Message flow 'FRED/default/flow2' is reported as stopped.
BIP1031I: Message flow 'FRED/exegrp2/flow1' is reported as stopped.
BIP8071I: Successful command completion.
```


This slide shows an example of stopping all message flows, in all execution groups, in the broker called FRED. Each message flow reported as stopped will be reported by this command.

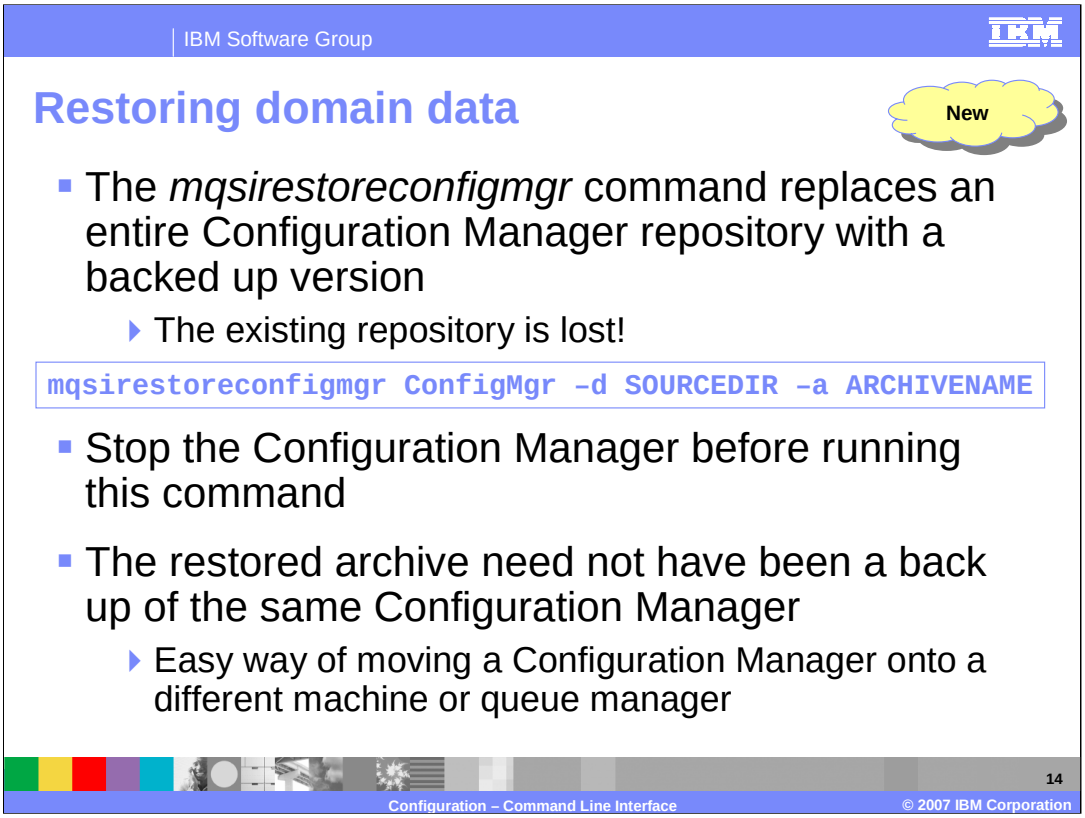
The command will wait up to 60 seconds to receive the response to the command.



As discussed earlier, the Configuration Manager does not now use an external DB2 database. Instead, it uses an internal database which is installed and configured automatically by the product installation process. However, the Configuration Manager repository contains useful information, and it may be desirable to back up the data contained here. For example, this may be useful when reporting problems to IBM; one of the pieces of diagnostic information may be a backup of this repository.

The repository can be backed up using the command shown on this slide. The –d parameter is used to specify where the backup will be stored. Before executing this command, ensure that the Configuration Manager has been stopped.

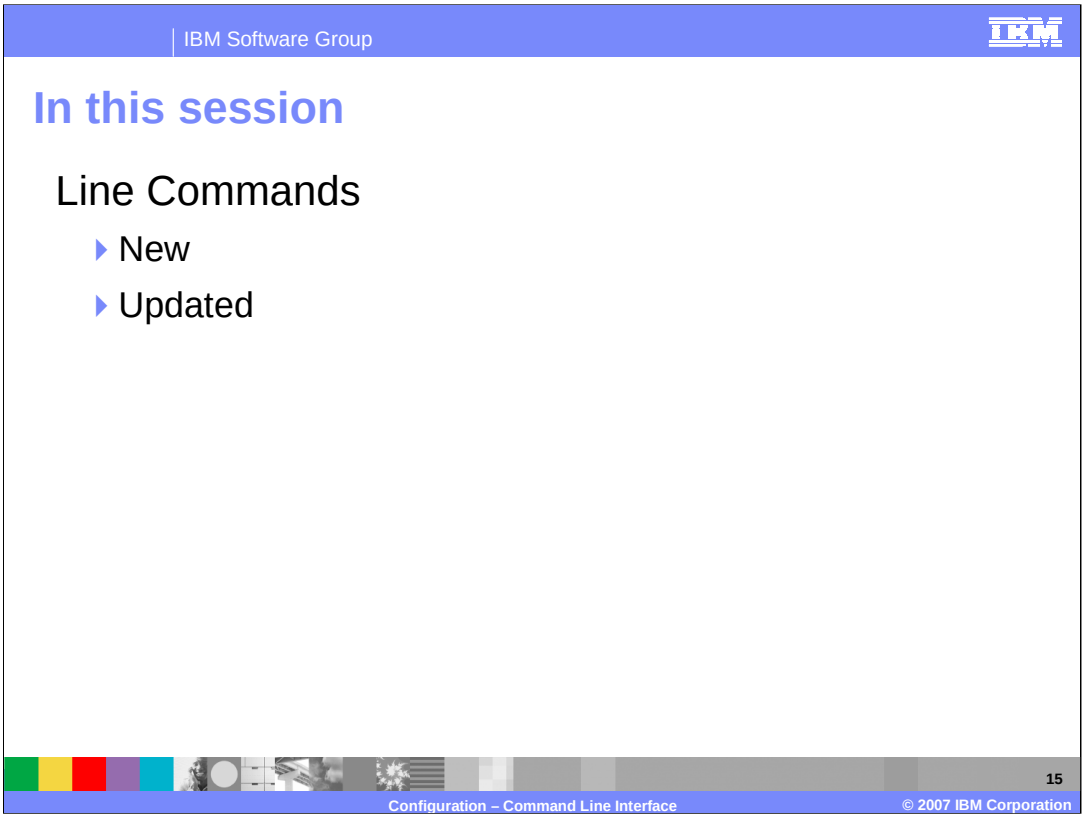
The backup file name will be generated, based on the name of the Configuration Manager. This can be changed by using the –a parameter.



Conversely, to restore the Configuration Manager from a backup, use the command shown here. Again, the Configuration Manager should be stopped before running this command.

It is important to note that this command will overwrite anything that is contained in the current Configuration Manager repository. Hence, you should be sure that you intend to perform this action before going ahead with the command.

The Configuration Manager to be restored does not have to exist. If it does not exist, this command will effectively create a new Configuration Manager, based on the contents of the backup. The ConfigMgr parameter on the command is used to create a Configuration Manager of that name. Hence, this command can be used to replicate Configuration Manager definitions to new instances.



This session covered some of the new and updated line commands in WebSphere Message Broker Version 6.

References

- WebSphere Message Broker library:

http://www-306.ibm.com/software/integration/wbimessagebroker/library/

- WebSphere Message Broker Information Center:

http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp

Trademarks, copyrights, and disclaimers