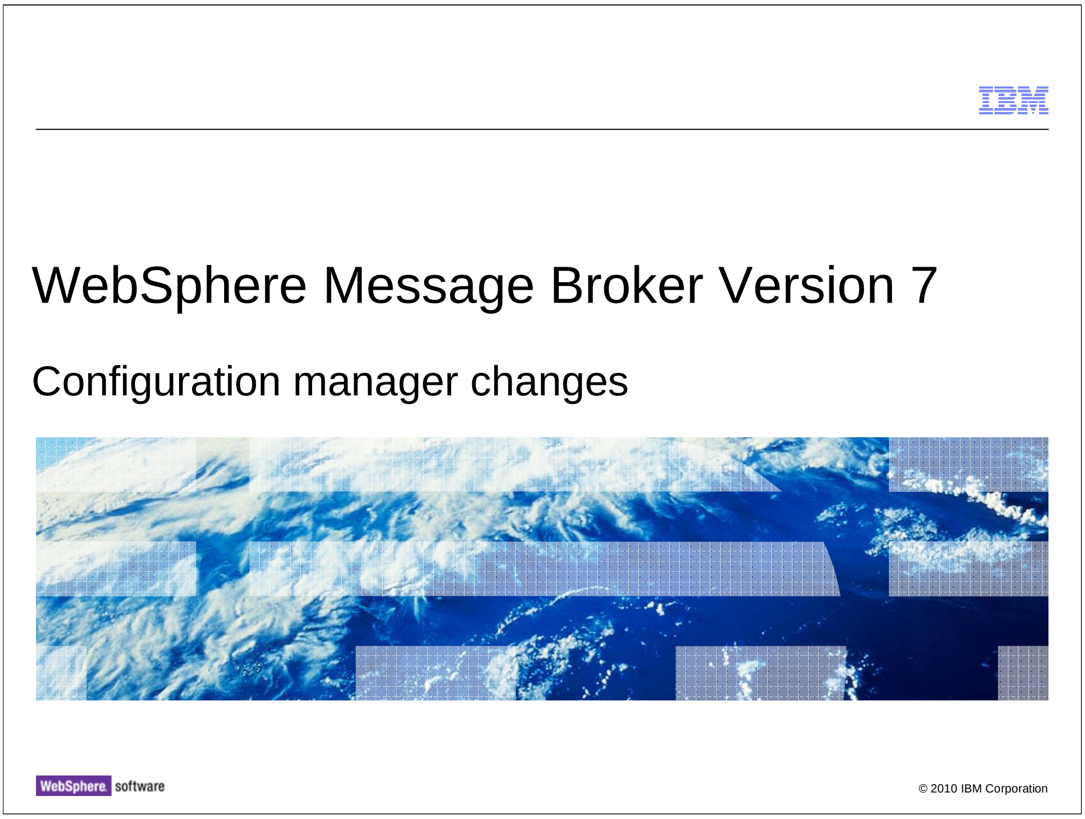
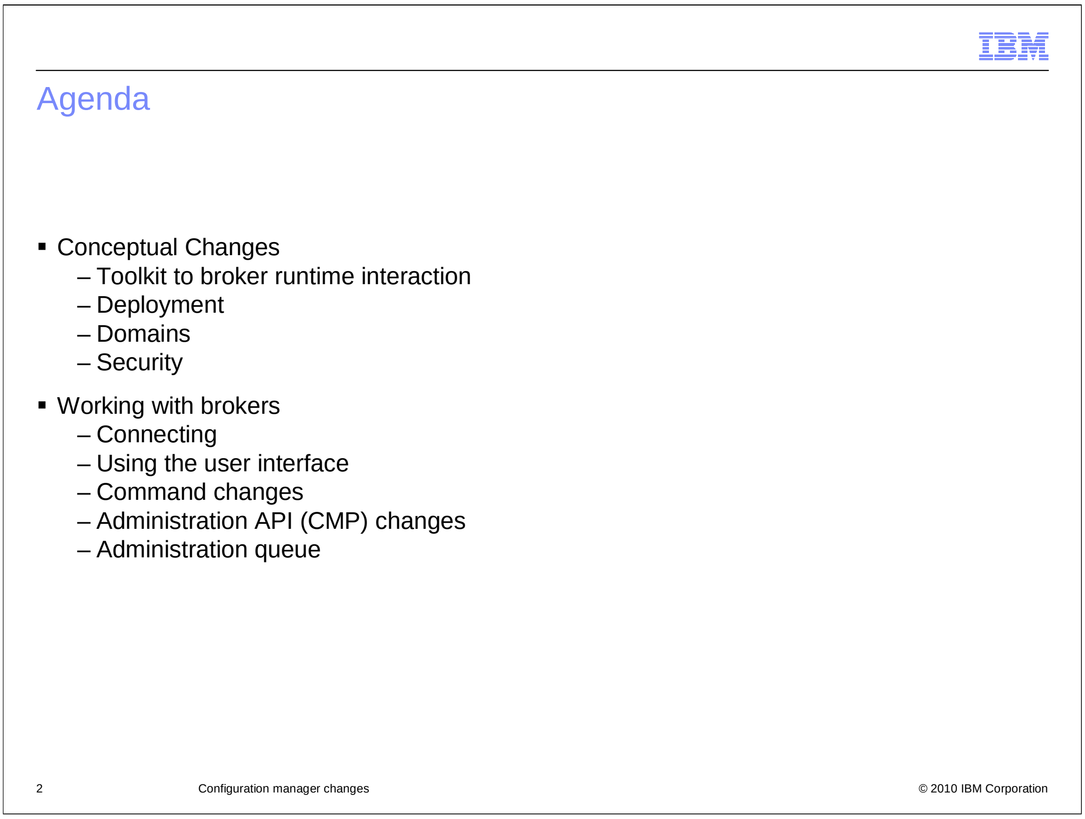


WebSphere Message Broker Version 7

Configuration manager changes

This session discusses the changes that have been made to the Configuration Manager in Message Broker version 7. As you will have seen in the Release Overview session, the configuration manager component has been removed in version 7. This has implications on the installation and configuration of the broker, and some of the facilities that were formerly provided by the configuration manager.

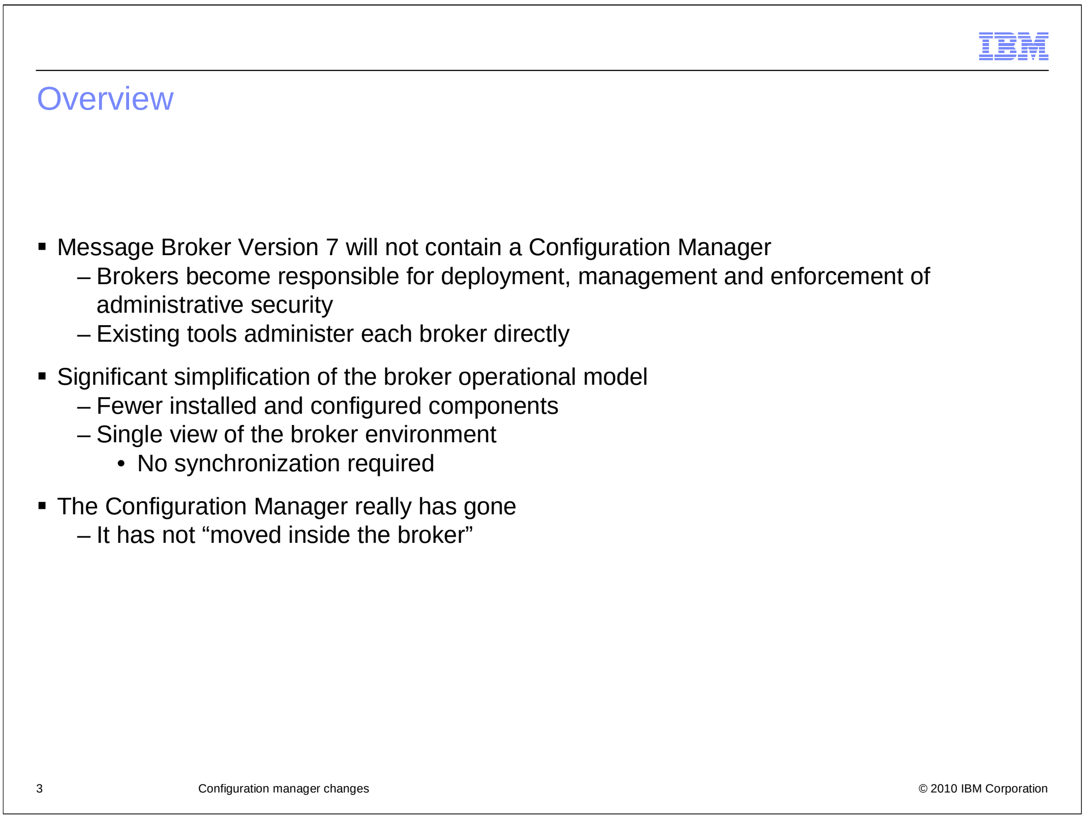


Agenda

- Conceptual Changes
  - Toolkit to broker runtime interaction
  - Deployment
  - Domains
  - Security
- Working with brokers
  - Connecting
  - Using the user interface
  - Command changes
  - Administration API (CMP) changes
  - Administration queue

This session will explain the nature of the changes made to the configuration manager, and the effect on the toolkit connections, and how to deploy and manage applications. The concept of domains has changed in version 7, as has the security mechanisms for administration.

Secondly, it will explain how to use the updated interfaces, including the toolkit and Message Broker Explorer, the command interface, and the administration API. It will also show how commands are logged and stored for audit purposes.



Overview

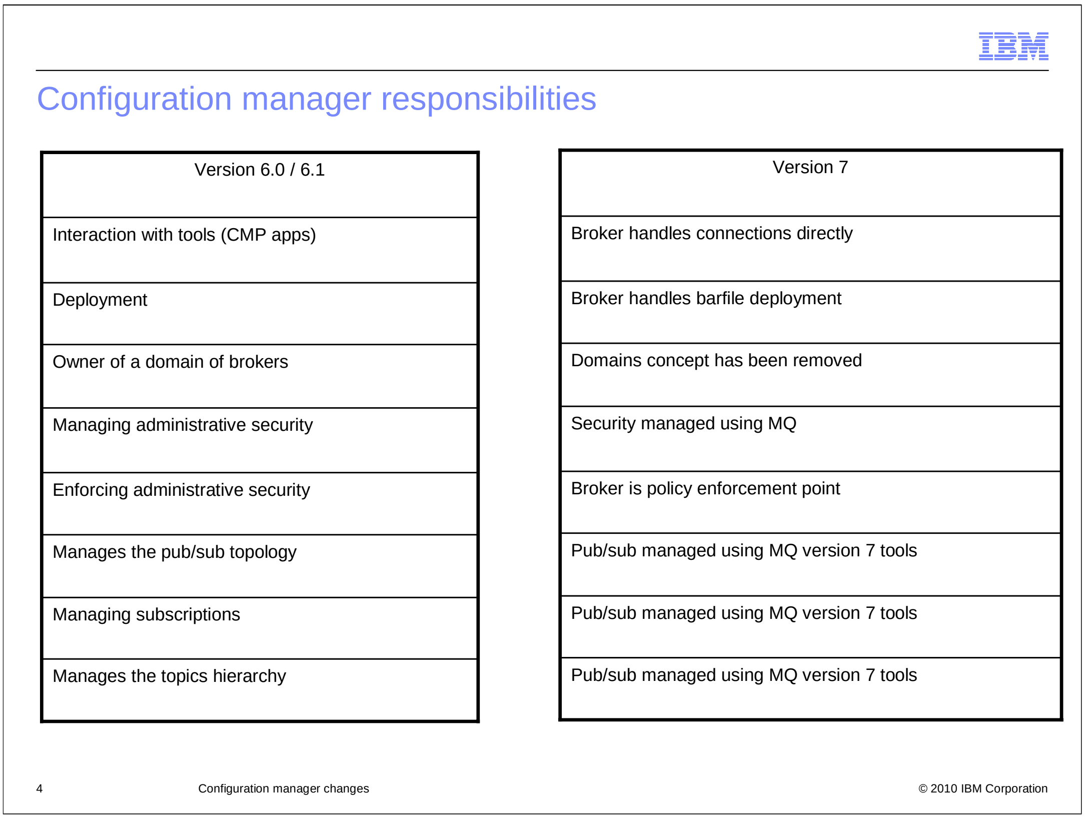
- Message Broker Version 7 will not contain a Configuration Manager
  - Brokers become responsible for deployment, management and enforcement of administrative security
  - Existing tools administer each broker directly
- Significant simplification of the broker operational model
  - Fewer installed and configured components
  - Single view of the broker environment
    - No synchronization required
- The Configuration Manager really has gone
  - It has not “moved inside the broker”

3 Configuration manager changes

In Message Broker version 7, the configuration manager has been completely removed. All the functions provided by the configuration manager in previous versions of Message Broker are now provided by the broker itself. All software components which formerly connected to the Configuration Manager, such as the Message Broker Toolkit, now connect directly to the target broker. This also includes the new Message Broker Explorer, the command interface, and any applications using the administration programming interface.

The result of this change is a much simpler operational model. If you are familiar with previous versions of Message Broker, you will see a much easier environment to install, configure and operate. A key consequence of this change is that there is now a single view of the runtime environment. In previous versions, the Configuration Manager did not always reflect a true picture of the deployed applications in the broker, and synchronization was occasionally an issue. In version 7, this scenario is impossible, and applications will always receive a true picture of the runtime environment.

You should also note that the functions of the Configuration Manager have been rewritten to run optimally inside the broker. The Configuration Manager has not been moved to run in a different environment. This means that connection times and deployment times are improved compared to previous versions.



Configuration manager responsibilities

Version 6.0 / 6.1	Version 7
Interaction with tools (CMP apps)	Broker handles connections directly
Deployment	Broker handles barfile deployment
Owner of a domain of brokers	Domains concept has been removed
Managing administrative security	Security managed using MQ
Enforcing administrative security	Broker is policy enforcement point
Manages the pub/sub topology	Pub/sub managed using MQ version 7 tools
Managing subscriptions	Pub/sub managed using MQ version 7 tools
Manages the topics hierarchy	Pub/sub managed using MQ version 7 tools

4 Configuration manager changes © 2010 IBM Corporation

This slide provides a comparison of the key Configuration Manager functions in version 6.1 to the equivalent function in version 7.

The first item is the interaction with the tools, such as the Message Broker Toolkit, and the Configuration Manager API. In version 7, all tools and applications using the API connect directly to the target broker. The broker handles these requests itself, and the connecting applications will need to specify the connection details of the broker, in exactly the same way that it specified the connection details of the Configuration Manager.

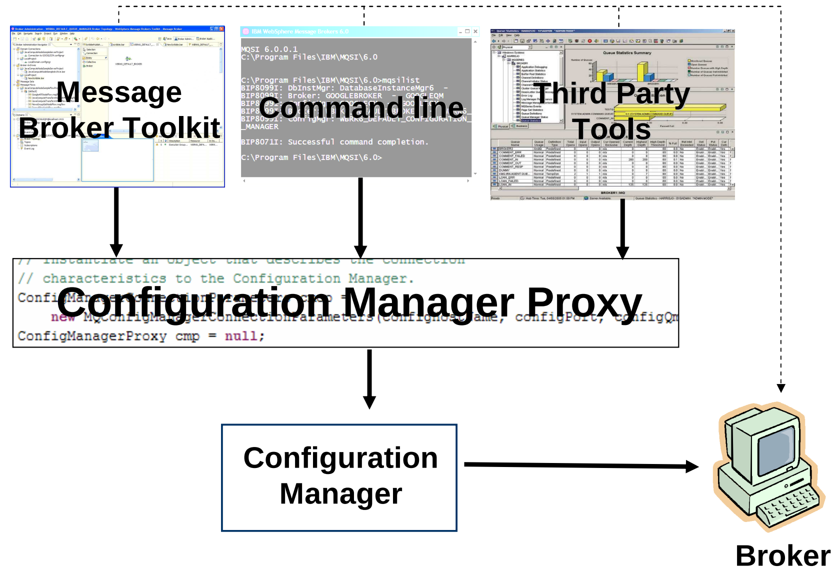
For the deployment of applications, the barfile is deployed directly into the broker, using either the supplied tools, or your own applications.

The domain concept was a direct consequence of the usage of the Configuration Manager. In some scenarios, a single Configuration Manager instance was used to manage several broker instances, and this grouping of brokers was known as a domain. With the removal of the Configuration Manager, the concept of the Message Broker domain has also been removed. However, version 7 has introduced the concept of broker sets, which is a tool to allow separate broker instances to be logically grouped, and displayed using the Message Broker Explorer.

One of the functions provided by the Configuration Manager was the administration security, and the creation and maintenance of access control lists for the broker. The administration point is now provided by the underlying MQ queue manager, although the broker remains the security policy enforcement point. A fuller description of this facility is given in another IBM Education Assistant session.

And finally, all topics concerned with publish / subscribe have now been completely replaced by the publish / subscribe tools in WebSphere MQ.

Interaction with tools – Version 6.0 / 6.1

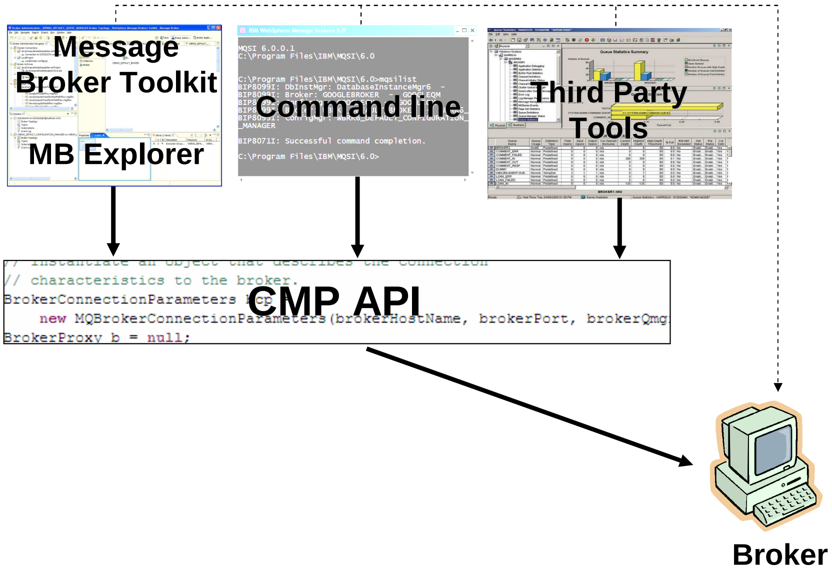


```
MQSI 6.0.0.1
C:\Program Files\IBM\MQSI\6.0

C:\Program Files\IBM\MQSI\6.0>mqsilist
BIP8099I: DbInstMgr: DatabaseInstanceMgr6  -
BIP8099I: Broker: GOOGLEBROKER  -  GOOGLEQM
BIP8099I: ConfigMgr: WBRK6_DEFAULT_CONFIGURATION
_MANAGER

BIP8071I: Successful command completion.

C:\Program Files\IBM\MQSI\6.0>
```

```
// instantiate an object that describes the connection
// characteristics to the Configuration Manager.
ConfigManagerConnectionParameters cmcp =
    new MQConfigManagerConnectionParameters(configHostName, configPort, configQm
ConfigManagerProxy cmp = null;
```

As a reminder, in previous versions of Message Broker, all applications connected to the broker using the Configuration Manager Proxy API. This included the Message Broker Toolkit and the command line interface, and any user-written Java applications which needed access to Configuration Manager function.

Interaction with tools - Version 7

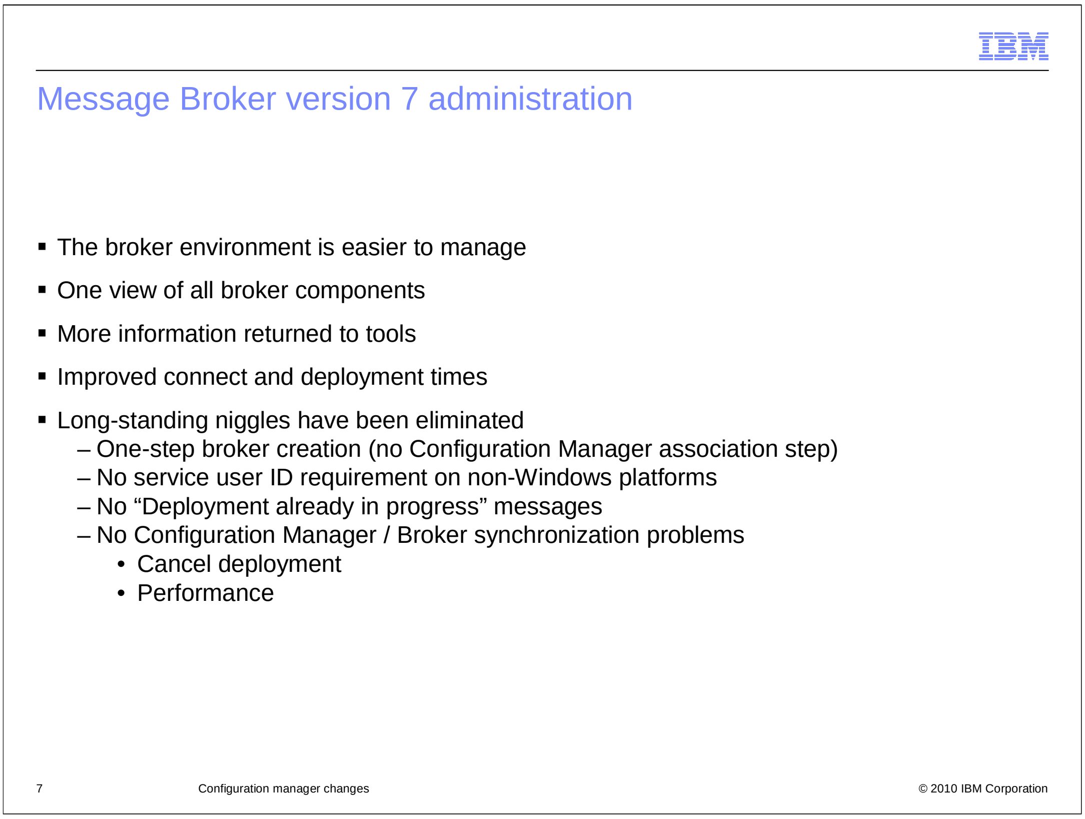


```
MQSI 6.0.0.1
C:\Program Files\IBM\MQSI\6.0

C:\Program Files\IBM\MQSI\6.0>mqsilist
BIP8099I: DbInstMgr: DatabaseInstanceMgr6  -
BIP8099I: Broker: GOOGLEBROKER  -  GOOGLEQM
BIP8[illegible]
BIP8[illegible]
BIP8099I: ConfigMgr: WBRK6_DEFAULT_CONFIGURATION_
_MANAGER

BIP8071I: Successful command completion.

C:\Program Files\IBM\MQSI\6.0>
```

```
// instantiate an object that describes the connection
// characteristics to the broker.
BrokerConnectionParameters bcp =
    new MQBrokerConnectionParameters(brokerHostName, brokerPort, brokerQmg
BrokerProxy b = null;
```

6 Configuration manager changes © 2010 IBM Corporation

In version 7, the same tools can invoke the same function, but in this case they connect directly to the target broker. The application interface is now known as the CMP API, although it operates in exactly the same way as it did in previous versions.



Message Broker version 7 administration

- The broker environment is easier to manage
- One view of all broker components
- More information returned to tools
- Improved connect and deployment times
- Long-standing niggles have been eliminated
  - One-step broker creation (no Configuration Manager association step)
  - No service user ID requirement on non-Windows platforms
  - No “Deployment already in progress” messages
  - No Configuration Manager / Broker synchronization problems
    - Cancel deployment
    - Performance

This slide discusses the main benefits that you will see as a result of the removal of the Configuration Manager.

First, the overall broker environment is easier to configure and manager, because there are fewer installed components. Secondly, all tools and applications will see a consistent view of the broker environment, and the deployed broker applications. There is no possibility of the brokers getting “out of sync” with the Configuration Manager.

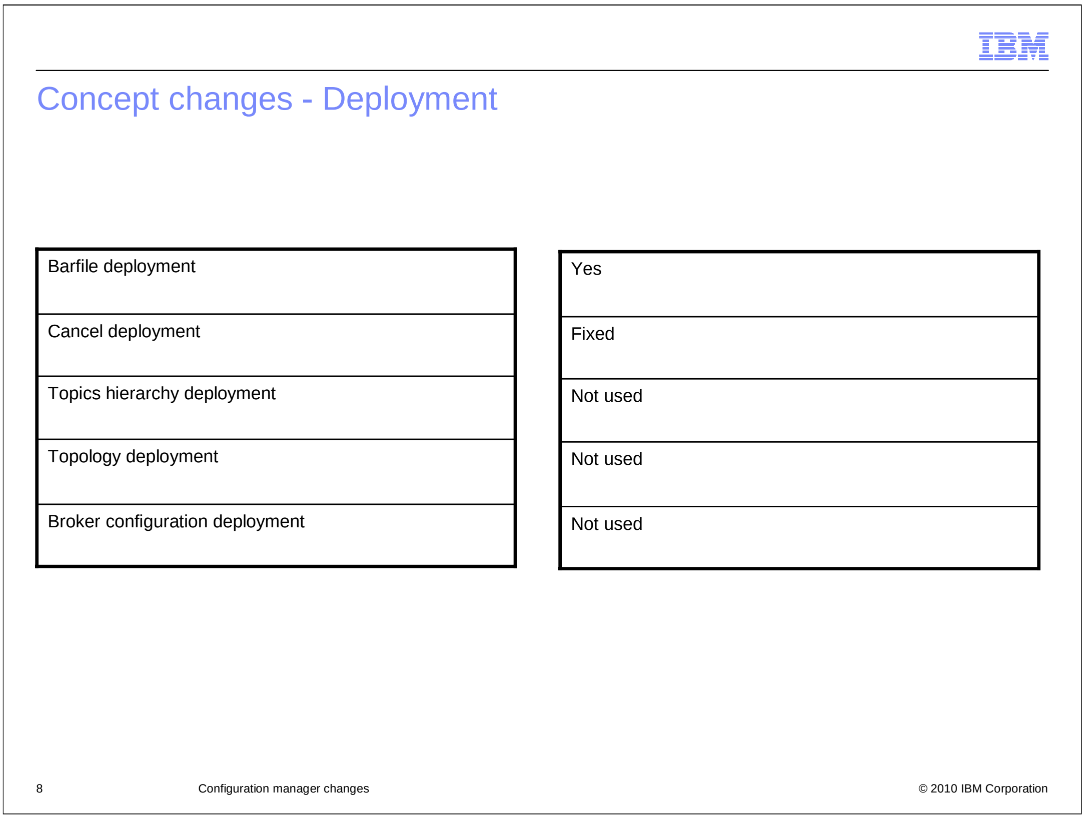
At the same time, the information provided by the broker about the broker environment, and about the deployed applications, has been significantly increased. This extra information is displayed in the Message Broker Toolkit and the Explorer, and is also available to user-written applications.

In some larger broker deployments, it was possible for the connection to the Configuration Manager to take a long time. Again, as a result of the removal of this component, connection times to the target brokers will be much faster than previous versions., allowing better control and management of deployed applications.

A number of long-standing requirements have also been addressed. First, the creation of a broker is now much simpler than previously, and is now a single-stop operation. A Toolkit user can create a broker, connect to the broker and create an execution group, and can immediately start using the broker for deployments. Note that a database is no longer required; this is discussed in the next IBM Education Assistant session.

Next, the requirement to use a service user ID and password has been removed on most platforms. On UNIX and Linux platforms, in previous versions, the broker ran under the user ID of the person who issues the broker start request. However, this is still required on Windows, because of the way that the Windows service manager is used.

It is now possible for multiple users to connect and deploy applications to the broker at the same time. This avoids the problem that sometimes occurred where users attempted to deploy applications simultaneously.



Concept changes - Deployment

Barfile deployment	Yes
Cancel deployment	Fixed
Topics hierarchy deployment	Not used
Topology deployment	Not used
Broker configuration deployment	Not used

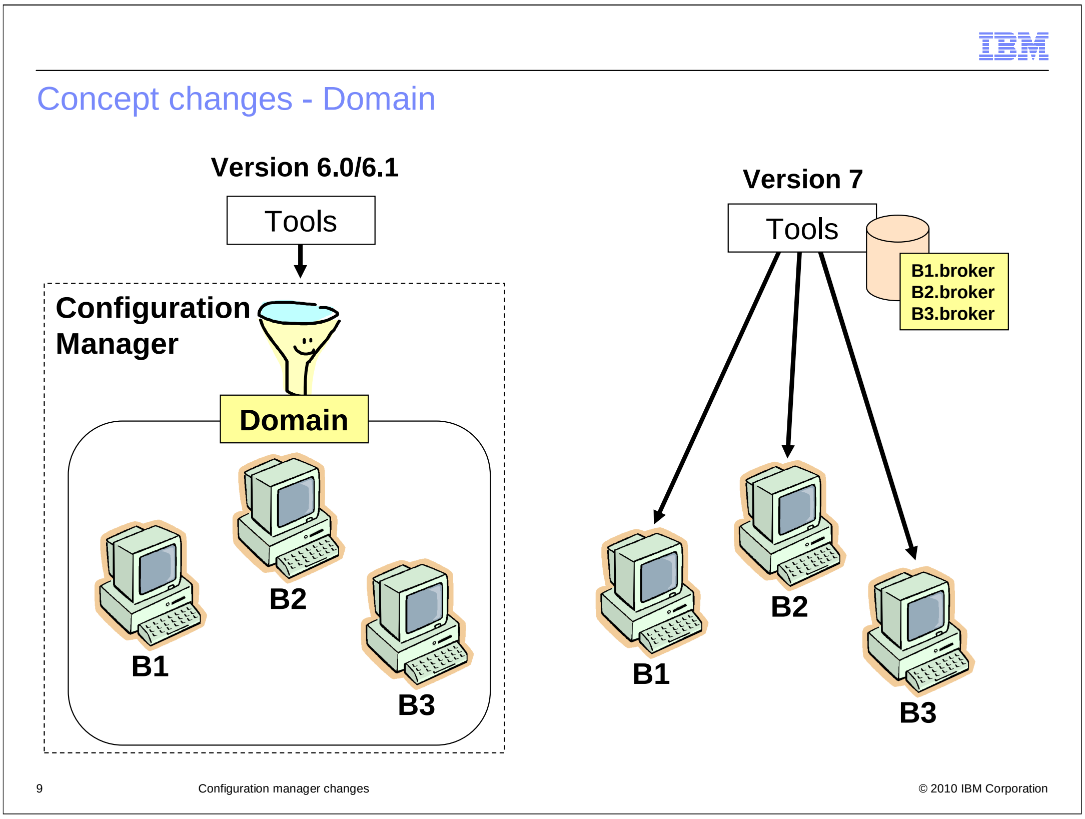
8 Configuration manager changes © 2010 IBM Corporation

This slide highlights the main differences in the area of deployment.

First, the barfile deployment process is unchanged from previous versions. The Toolkit, MB Explorer, command interface or user-written application, will all deploy a barfile in exactly the same way to the target broker.

If you want to cancel a deployment, the process has been changed to avoid the possibility of affecting other users of the broker.

The other items of topics hierarchy, topology deployment and broker configuration, are not required in version 7. All actions that are taken to define or update the broker topology are performed directly on the brokers themselves. Any change of broker parameters are actioned immediately, and there is no requirement to reflect this change in the broker topology definition.



Concept changes - Domain

Version 6.0/6.1

Tools

Configuration Manager

Domain

B1

B2

B3

Version 7

Tools

B1.broker
B2.broker
B3.broker

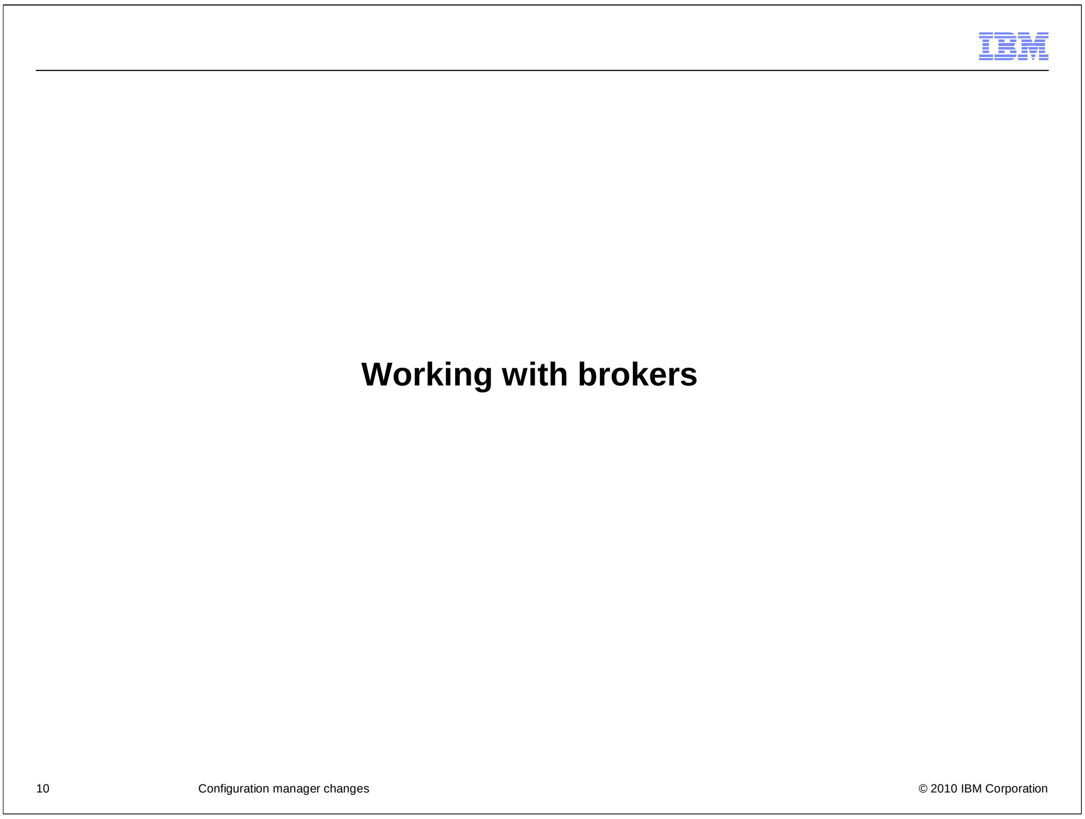
B1

B2

B3

9 Configuration manager changes © 2010 IBM Corporation

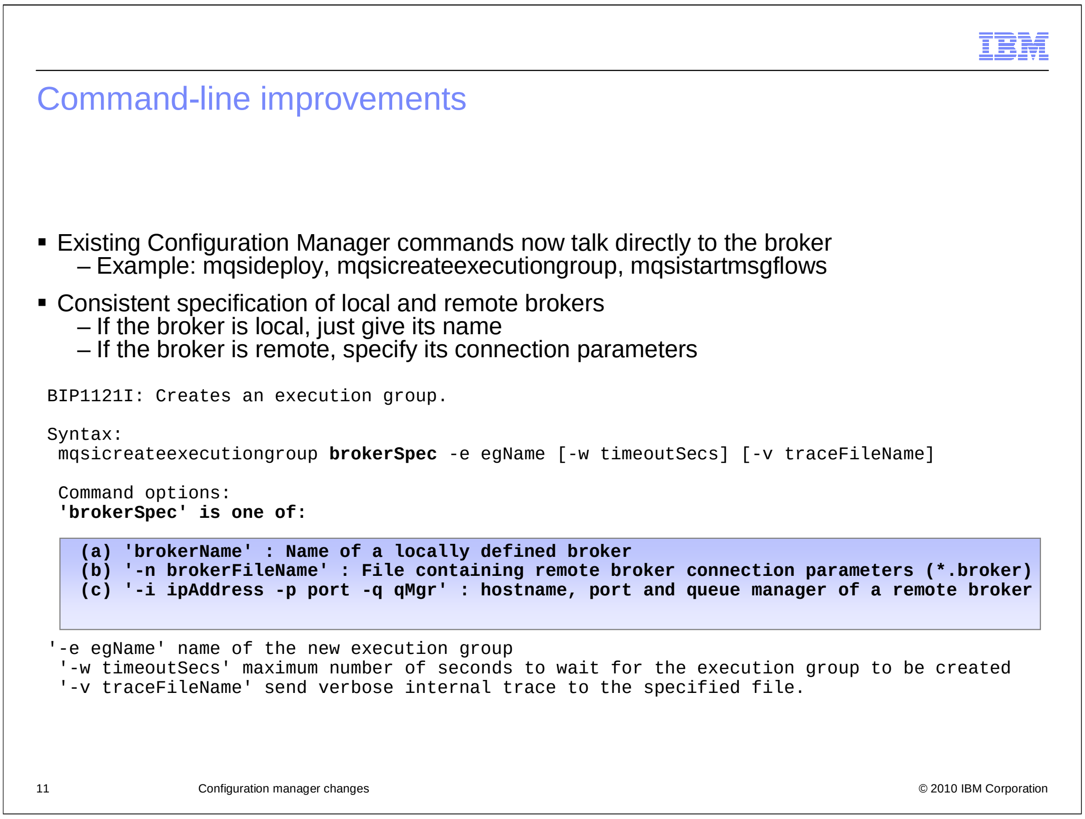
In summary, this slide shows the differences between Message Broker version 6.1 and Message Broker version 7. In version 7, the Configuration Manager has been removed completely, and consequently the concept of a broker domain no longer applies. In version 7, the brokers are managed independently, and all tools, such as the Message Broker Toolkit and the Message Broker Explorer, connect directly to all defined brokers.



Working with brokers

10 Configuration manager changes © 2010 IBM Corporation

This session now moves on to describe how to work with brokers, and how this has changed in Message Broker version 7.



Command-line improvements

- Existing Configuration Manager commands now talk directly to the broker
  - Example: mqsideploy, mqsicreateexecutiongroup, mqsistartmsgflows
- Consistent specification of local and remote brokers
  - If the broker is local, just give its name
  - If the broker is remote, specify its connection parameters

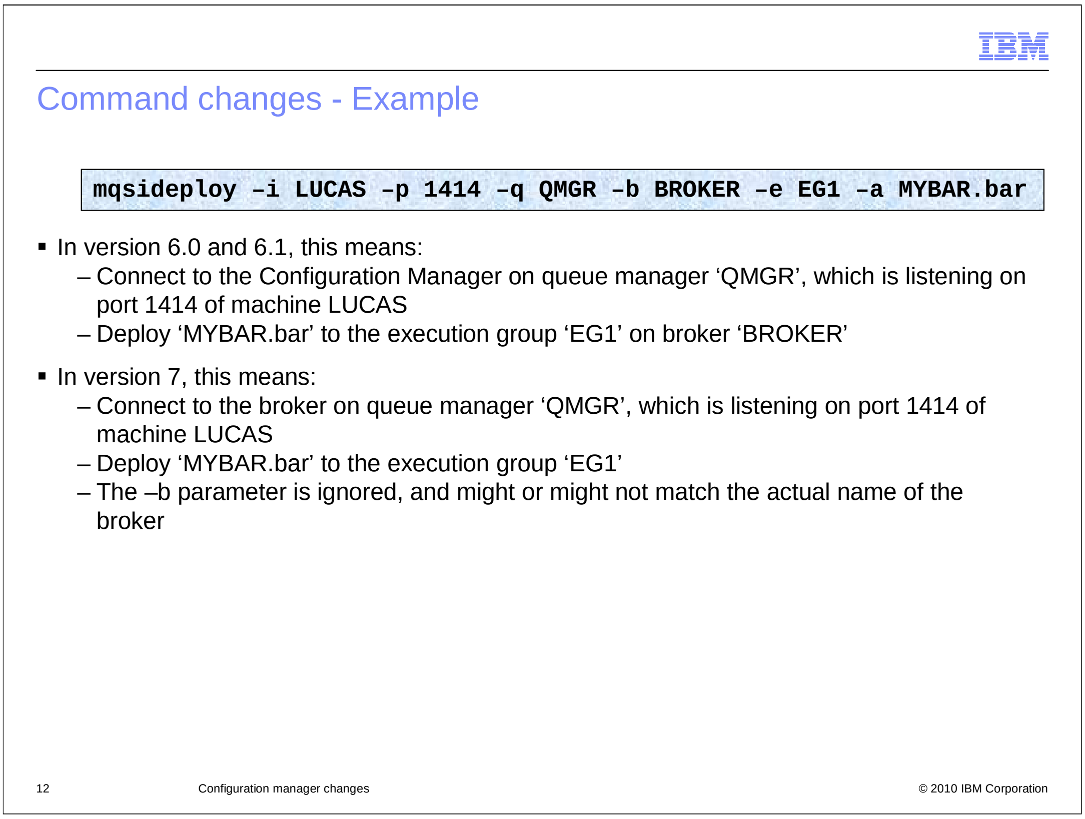
```
BIP1121I: Creates an execution group.

Syntax:
 mqsicreateexecutiongroup brokerSpec -e egName [-w timeoutSecs] [-v traceFileName]

 Command options:
 'brokerSpec' is one of:

   (a) 'brokerName' : Name of a locally defined broker
   (b) '-n brokerFileName' : File containing remote broker connection parameters (*.broker)
   (c) '-i ipAddress -p port -q qMgr' : hostname, port and queue manager of a remote broker

'-e egName' name of the new execution group
 '-w timeoutSecs' maximum number of seconds to wait for the execution group to be created
 '-v traceFileName' send verbose internal trace to the specified file.
```

11 Configuration manager changes © 2010 IBM Corporation

In previous versions, there were several commands which communicated with the Configuration Manager, which in turn communicated with the required broker. Deploying a barfile, creating an execution group, and starting a message flow, are some examples of these types of commands.

These commands still exist in Message Broker version 7, but they now communicate directly with the broker. As in previous versions of Message Broker, these commands sometimes require specific connection details such as tcp/ip connection details, and queue manager information.

If the broker is a local broker, running on the same system as the command line application, then you only need to specify the name of the required broker. If the broker is remote, then you must specify the tcp/ip and queue manager connection details, using either the definitions directly, or contained within a file. An example is shown on this slide.



Command changes - Example

```
mqsideploy –i LUCAS –p 1414 –q QMGR –b BROKER –e EG1 –a MYBAR.bar
```

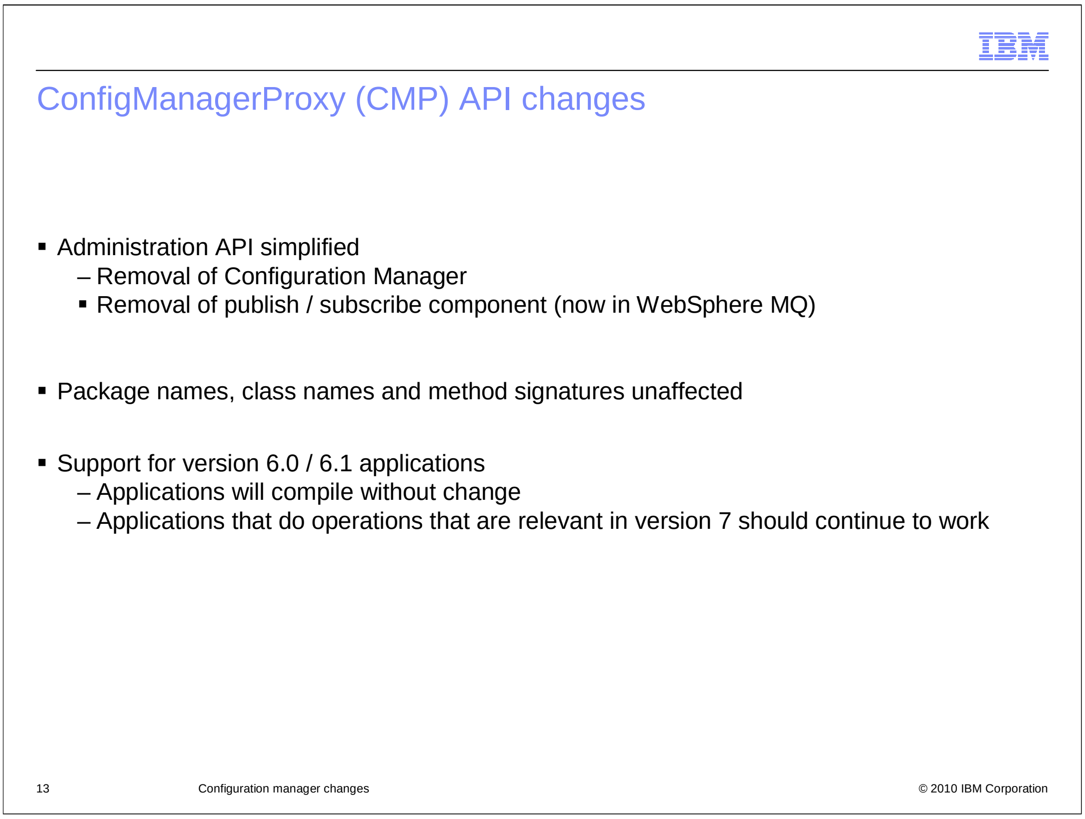
- In version 6.0 and 6.1, this means:
  - Connect to the Configuration Manager on queue manager ‘QMGR’, which is listening on port 1414 of machine LUCAS
  - Deploy ‘MYBAR.bar’ to the execution group ‘EG1’ on broker ‘BROKER’
- In version 7, this means:
  - Connect to the broker on queue manager ‘QMGR’, which is listening on port 1414 of machine LUCAS
  - Deploy ‘MYBAR.bar’ to the execution group ‘EG1’
  - The –b parameter is ignored, and might or might not match the actual name of the broker

12 Configuration manager changes © 2010 IBM Corporation

This slide shows an example of performing a deployment, using the “mqsi deploy” command.

In version 6 and 6.1, this command required the specification of the “-b BROKER” parameter. This was because the Configuration Manager managed several brokers, and specification of the TCP/IP address and queue manager did not provide sufficient information to allow the barfile to be deployed. The queue manager name specified by the “-q” parameter was the name of the queue manager used by the Configuration Manager.

In version 7, since the command interface connects directly to the broker, the “-q” parameter is the queue manager of the broker itself. The “-b BROKER” parameter is no longer required, and is ignored. If your broker and Configuration Manager used the same queue manager in version 6.1, then the same command will work unchanged in version 7. If your broker used a separate queue manager, then you must change the connections details of these commands, to ensure that the command uses the connection details of the broker queue manager.



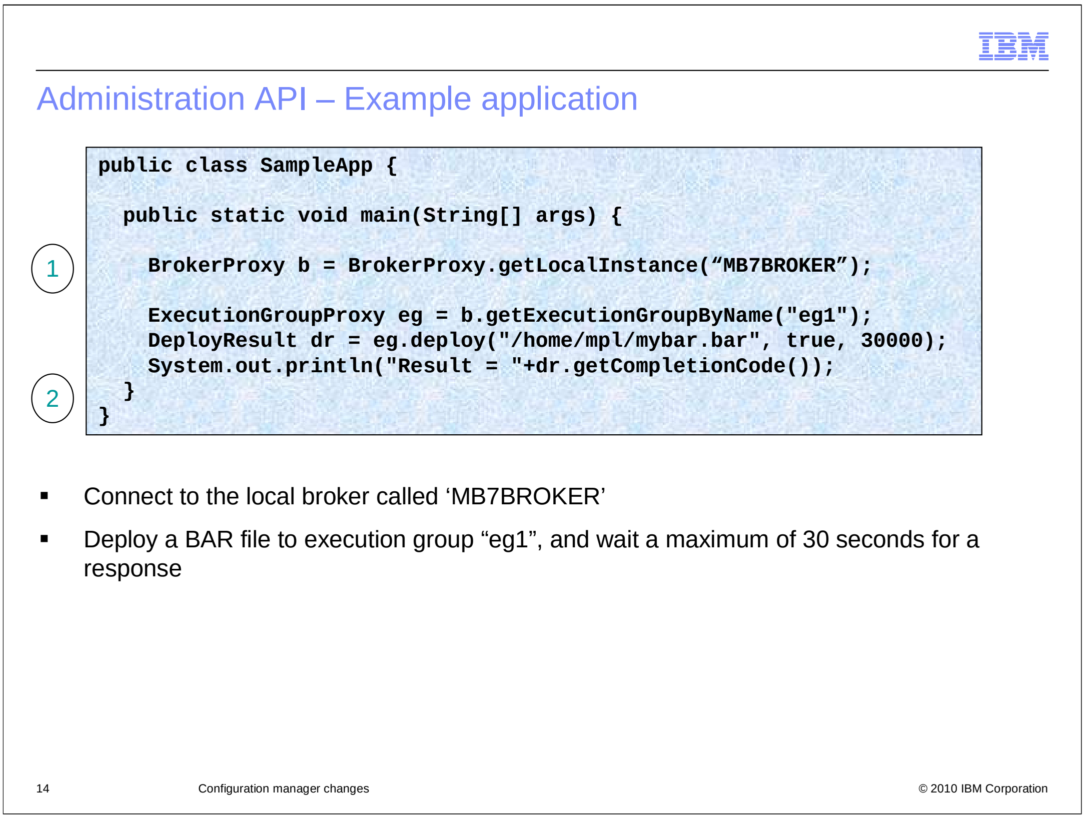
ConfigManagerProxy (CMP) API changes

- Administration API simplified
  - – Removal of Configuration Manager
    - Removal of publish / subscribe component (now in WebSphere MQ)
- Package names, class names and method signatures unaffected
- Support for version 6.0 / 6.1 applications
  - – Applications will compile without change
  - – Applications that do operations that are relevant in version 7 should continue to work

13 Configuration manager changes © 2010 IBM Corporation

In Message Broker version 7, the Configuration Manager Proxy API has been simplified. All API function concerned with the Configuration Manager, the broker topologies and the publish / subscribe hierarchy have been removed, since this function is no longer relevant.

Existing applications written for version 6 or 6.1 will continue to compile without change. However, the compile might produce warnings about deprecated functions, and you should consider cleaning up those applications to avoid these warnings.



Administration API – Example application

```
public class SampleApp {

  public static void main(String[] args) {

    BrokerProxy b = BrokerProxy.getLocalInstance(“MB7BROKER”);

    ExecutionGroupProxy eg = b.getExecutionGroupByName("eg1");
    DeployResult dr = eg.deploy("/home/mpl/mybar.bar", true, 30000);
    System.out.println("Result = "+dr.getCompletionCode());
  }
}
```

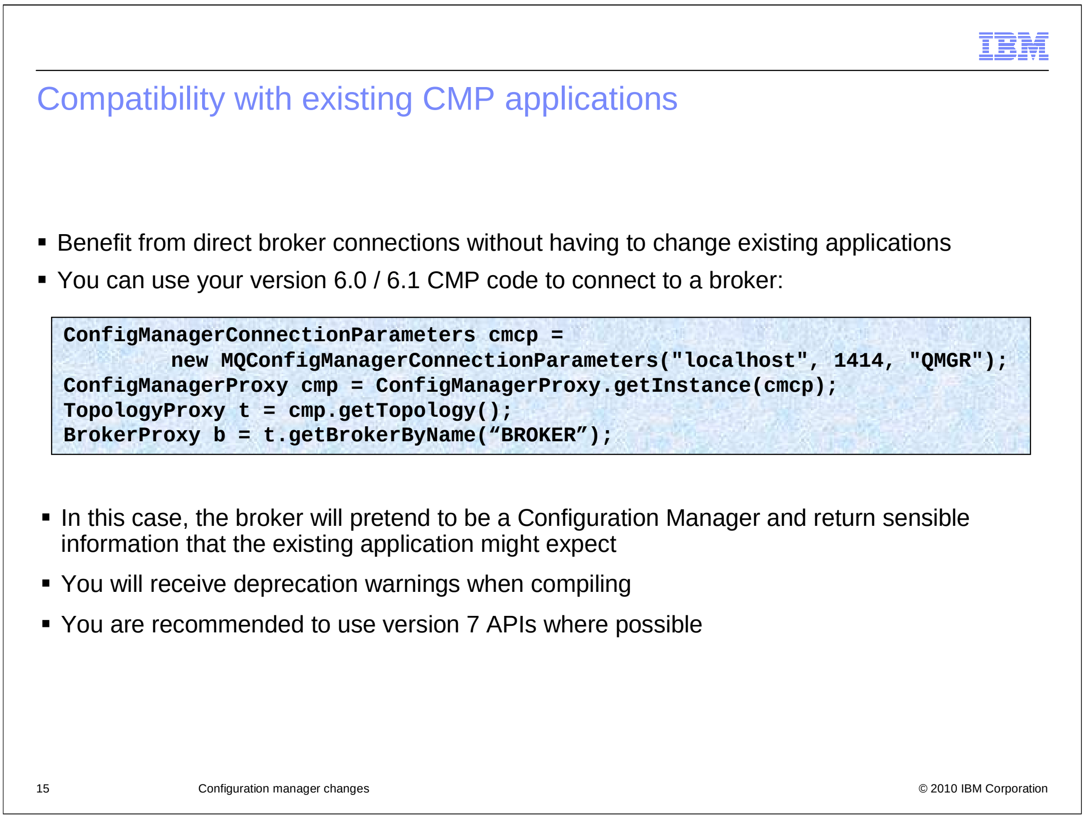
- Connect to the local broker called ‘MB7BROKER’
- Deploy a BAR file to execution group “eg1”, and wait a maximum of 30 seconds for a response

14 Configuration manager changes © 2010 IBM Corporation

This example illustrates the simplifications in version 7, and how the removal of the Configuration Manager has allowed user-written applications to take advantage of this.

In this case, the application is connecting to a local broker called MB7BROKER. There is no need to specify any additional connection details. The application connects to the broker, and returns a proxy object known as “b”. The application then uses this to retrieve details of the required execution group, “eg1”, and sets the proxy object “eg”.

Finally, this information is then used to deploy the barfile directly to the execution group.



Compatibility with existing CMP applications

- Benefit from direct broker connections without having to change existing applications
- You can use your version 6.0 / 6.1 CMP code to connect to a broker:

```
ConfigManagerConnectionParameters cmcp =
         new MQConfigManagerConnectionParameters("localhost", 1414, "QMGR");
ConfigManagerProxy cmp = ConfigManagerProxy.getInstance(cmcp);
TopologyProxy t = cmp.getTopology();
BrokerProxy b = t.getBrokerByName(“BROKER”);
```

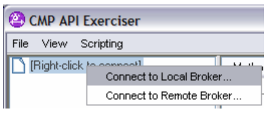
- In this case, the broker will pretend to be a Configuration Manager and return sensible information that the existing application might expect
- You will receive deprecation warnings when compiling
- You are recommended to use version 7 APIs where possible

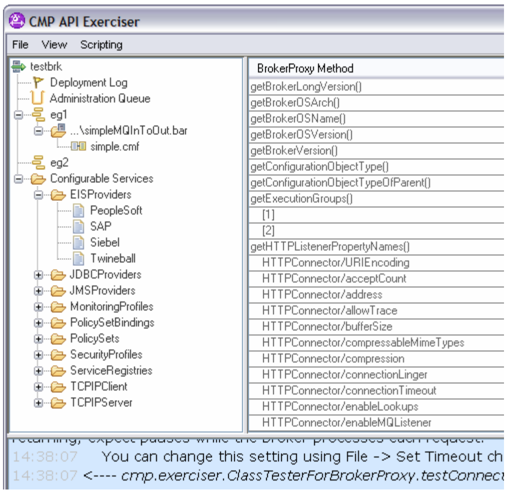
15 Configuration manager changes © 2010 IBM Corporation

If an application is using the more complex form of connection details, and has been migrated from Message Broker version 6 or 6.1, this will continue to work in version 7. However, it will connect directly to the broker. You will need to check that the connection details are correct for the version 7 environment.

The example shown here connects to the Configuration Manager, and obtains details of the broker runtime. This application will generated deprecation warnings when compiled, but will continue to run unchanged in the version 7 environment.

API exerciser changes





BrokerProxy Method
getBrokerLongVersion()
getBrokerOSArch()
getBrokerOSName()
getBrokerOSVersion()
getBrokerVersion()
getConfigurationObjectType()
getConfigurationObjectTypeOfParent()
getExecutionGroups()
[1]
[2]
getHTTPListenerPropertyNames()
HTTPConnector/URIEncoding
HTTPConnector/acceptCount
HTTPConnector/address
HTTPConnector/allowTrace
HTTPConnector/bufferSize
HTTPConnector/compressableMimeTypes
HTTPConnector/compression
HTTPConnector/connectionLinger
HTTPConnector/connectionTimeout
HTTPConnector/enableLookups
HTTPConnector/enableMQListener

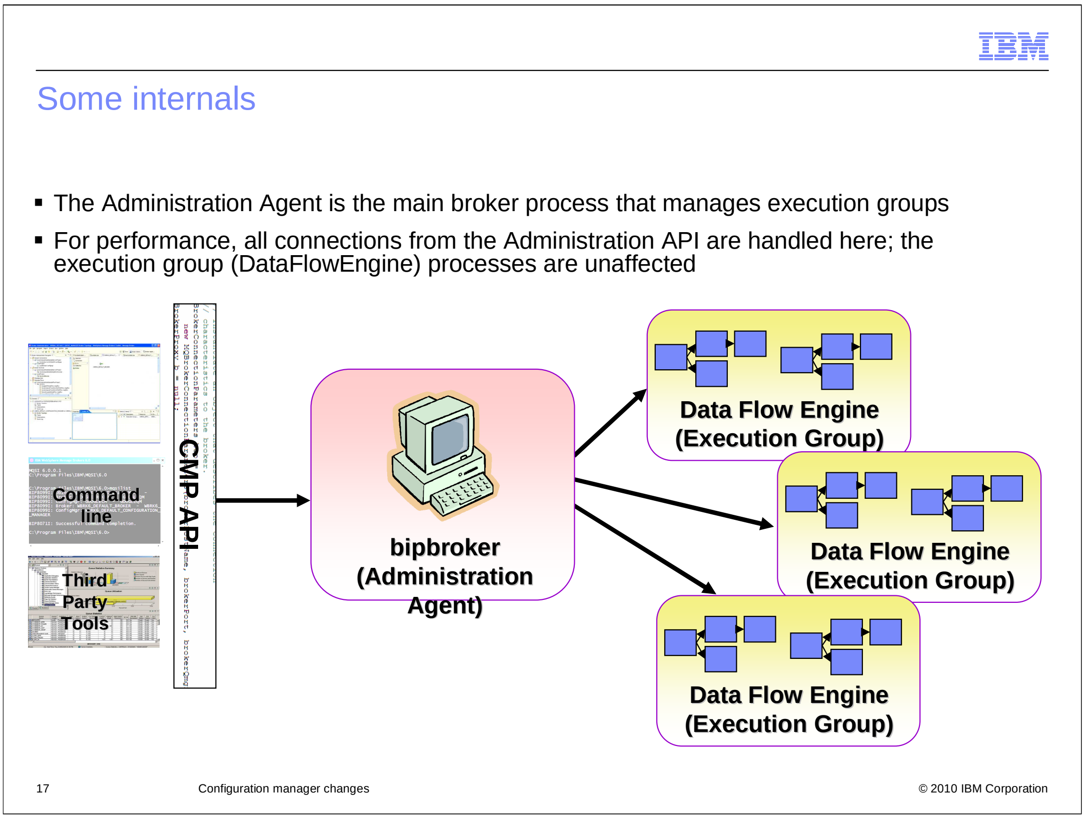
```
returning, expect pauses while the broker processes each request.
14:38:07     You can change this setting using File -> Set Timeout ch
14:38:07 <---- cmp.exerciser.ClassTesterForBrokerProxy.testConnect
```

- New actions
  - "Connect to Broker" option
  - View menu (Basic/Advanced/Everything)
  - Show barfiles under execution group; view and re-deploy them
  - Clipboard interaction - added 'Copy' option to table cells
  - Control service trace for execution group and message flow
- Removed all publish /subscribe administration options, including topology
- Added nodes, barfiles, message flow dependencies and icons to tree
- Added Administration Queue
- Added Configurable Services
- Right-click now uses sub-menus to group similar tasks

The API Exerciser has been updated to take account of the Configuration Manager changes in version 7. As a reminder, the API Exerciser is a standard component of the Message Broker product, providing a graphical user interface to the administration API, allowing connections to any defined brokers. All API functions are available, and full security is applied to users of this application. If changes are made to the API, then corresponding changes are made to the API Exerciser, as an aid to illustrating how this function can be used in your own applications.

For example, in version 7, the API Exerciser now distinguishes between local and remote brokers. If you select a local broker, you only need to select the required broker from the available list. New options such as the administration queue and configurable services have been added to the information presented.

The bulleted list shown on this slide provides a summary of the enhancements made to the API Exerciser.



Some internals

- The Administration Agent is the main broker process that manages execution groups
- For performance, all connections from the Administration API are handled here; the execution group (DataFlowEngine) processes are unaffected

Command line
Third Party Tools
CMP API
bipbroker (Administration Agent)
Data Flow Engine (Execution Group)
Data Flow Engine (Execution Group)
Data Flow Engine (Execution Group)

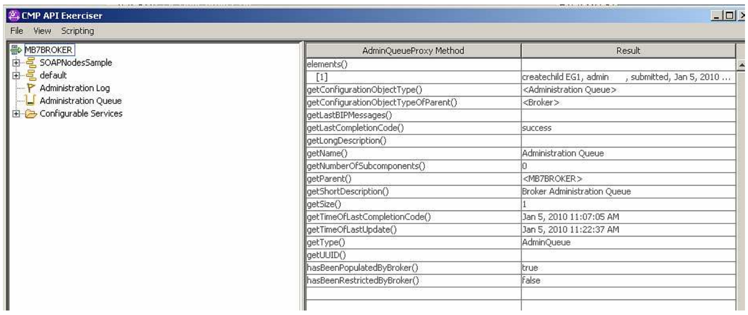
17 Configuration manager changes © 2010 IBM Corporation

If you are familiar with previous versions of Message Broker, you might be aware of the “bip broker” *(pronounced BIP)* process, otherwise known as the “Administration Agent” process. This process was used by the Configuration Manager to communicate with the broker execution groups. The Message Broker tools and user applications used the proxy API to communicate with the Configuration Manager, which in turn connected to the “bip broker” process.

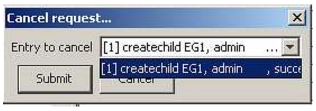
In version 7, the “bip broker” process is still used, but is accessed directly by applications and tools which use the administration API. The broker execution groups are still accessed using the “bip broker” process, not directly by user applications.

Administration queue

- What is the broker doing?
- What administration requests are outstanding?
- Allows work items not yet started to be cancelled
- Visualized in the API Exerciser and the Message Broker Explorer



AdminQueueProxy Method	Result
elements()	
[1]	createchild EG1, admin , submitted, Jan 5, 2010 ...
getConfigurationObjectType()	<Administration Queue>
getConfigurationObjectTypeOfParent()	<Broker>
getLastBIPMessages()	
getLastCompletionCode()	success
getLongDescription()	
getName()	Administration Queue
getNumberOfSubcomponents()	0
getParent()	<MB7BROKER>
getShortDescription()	Broker Administration Queue
getSize()	1
getTimeOfLastCompletionCode()	Jan 5, 2010 11:07:05 AM
getTimeOfLastUpdate()	Jan 5, 2010 11:22:37 AM
getType()	AdminQueue
getUUID()	
hasBeenPopulatedByBroker()	true
hasBeenRestrictedByBroker()	false



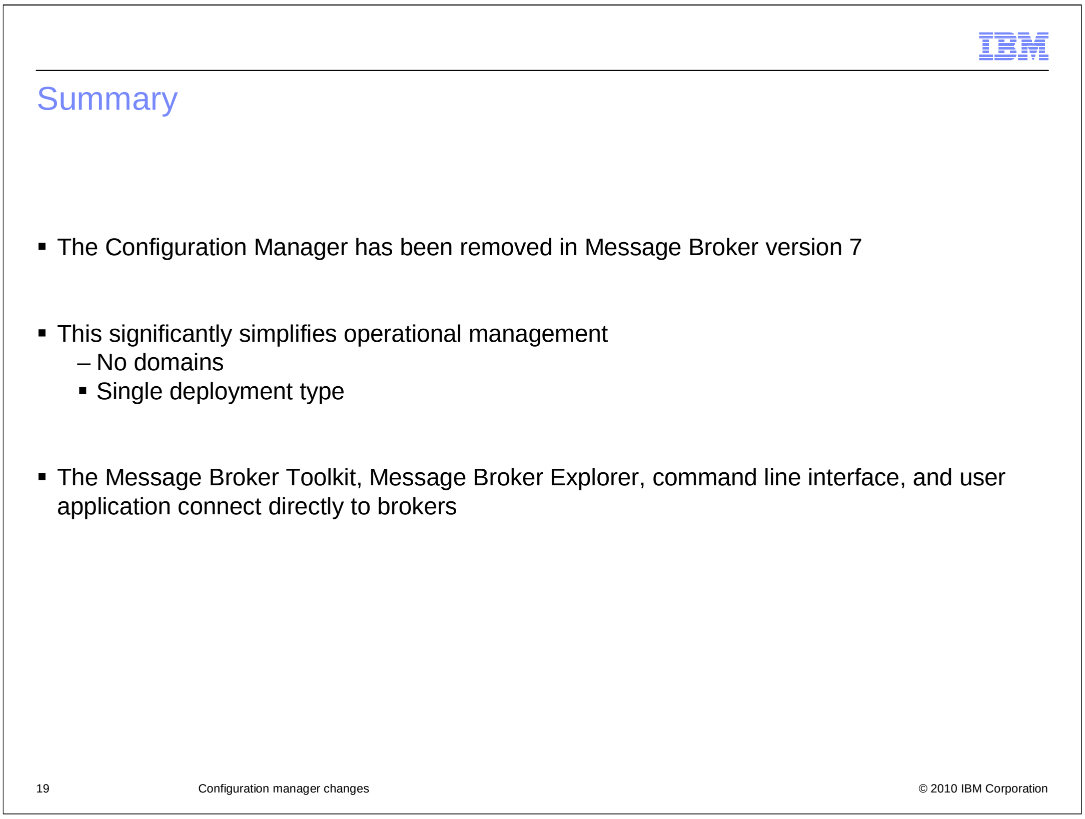
Entry to cancel: [1] createchild EG1, admin ...

[1] createchild EG1, admin , succe

In previous versions of Message Broker, it was possible to cancel an outstanding barfile deployment. This was sometimes necessary to clear the deployment queue to resolve a performance problem with the Configuration Manager. The cancel function did sometimes result in other problems, and with the removal of the Configuration Manager, this function has been replaced with an alternative mechanism to remove unfulfilled deployments.

In version 7, all requests to the broker are now managed using the Administration Queue. This queue is accessible using the Administration API, and is shown in the API Exerciser, and the Message Broker Explorer. The example shown here is the creation of an execution group, called EG1. This action has placed a small item in the administration queue icon. As additional requests are passed to the broker, the queue of items shown will increase. To display details of the outstanding requests, right-click the administration queue and select "Display outstanding requests". To cancel a particular request, select "Cancel request", and select the particular request.

This function is also available using the Message Broker Explorer.

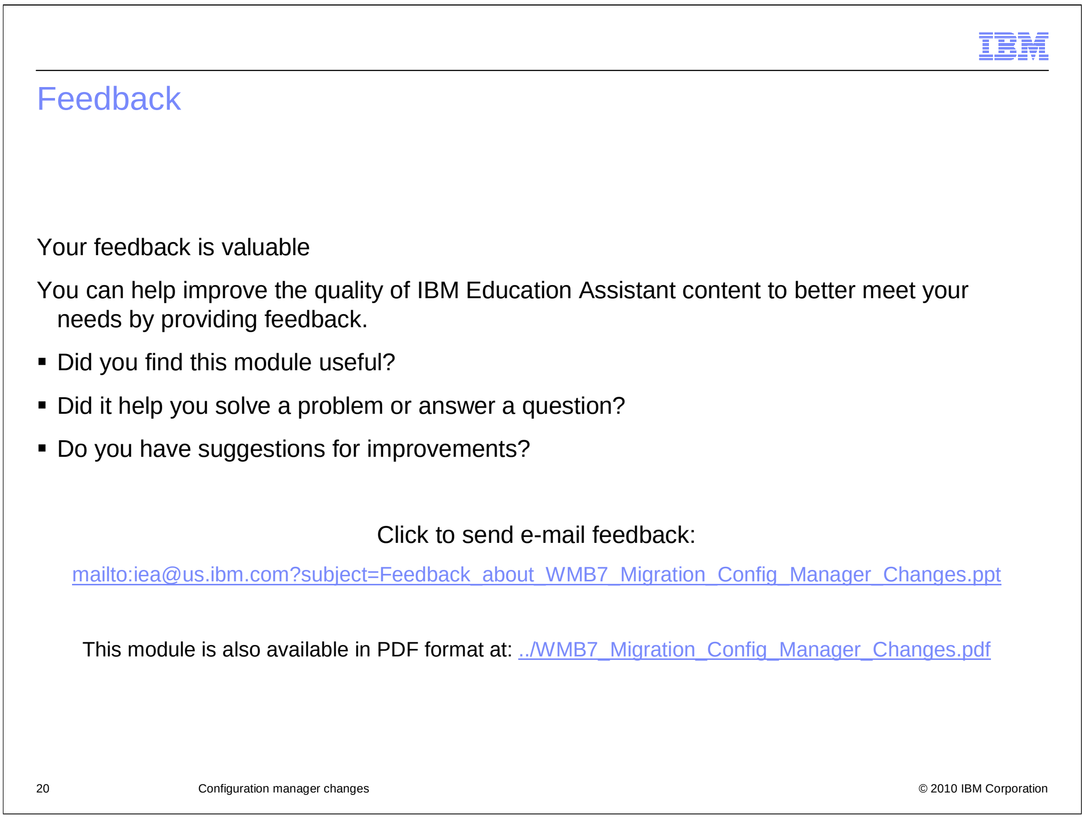


Summary

- The Configuration Manager has been removed in Message Broker version 7
- This significantly simplifies operational management
  - – No domains
    - Single deployment type
- The Message Broker Toolkit, Message Broker Explorer, command line interface, and user application connect directly to brokers

19 Configuration manager changes © 2010 IBM Corporation

In summary, the Configuration manager has been removed in Message Broker version 7. This has resulted in a significant simplification of the deployment architecture for Message Broker, and the concept of the broker domain has correspondingly been removed. The supplied tools such as the Message Broker Toolkit and Explorer, the command line interface and the administration API, all connect directly to the required broker.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

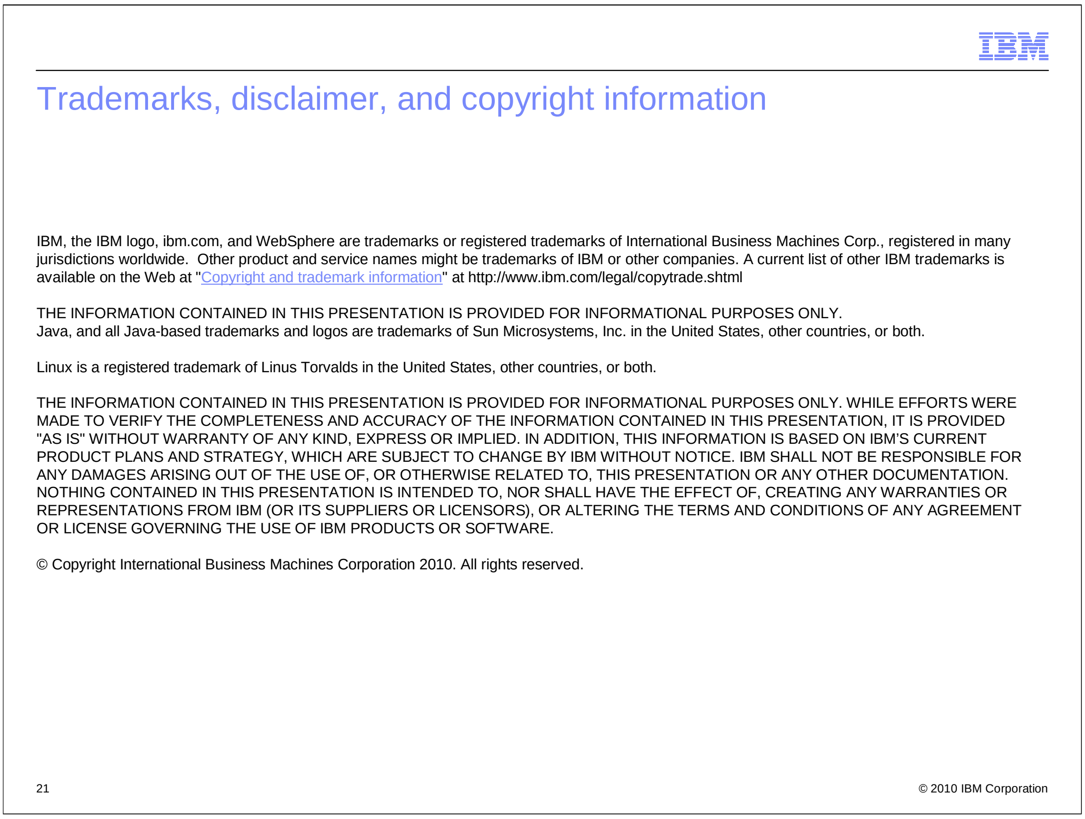
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WMB7_Migration_Config_Manager_Changes.ppt

This module is also available in PDF format at: ../WMB7_Migration_Config_Manager_Changes.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, disclaimer, and copyright information

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.