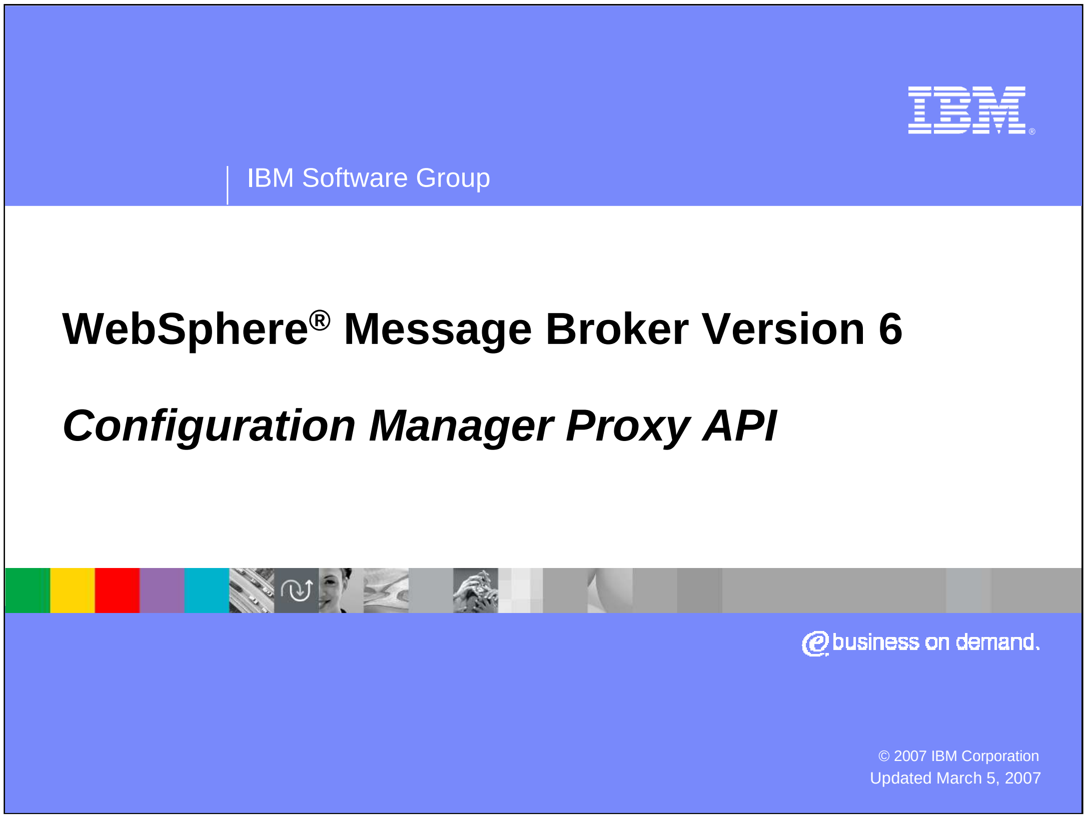


IBM Software Group

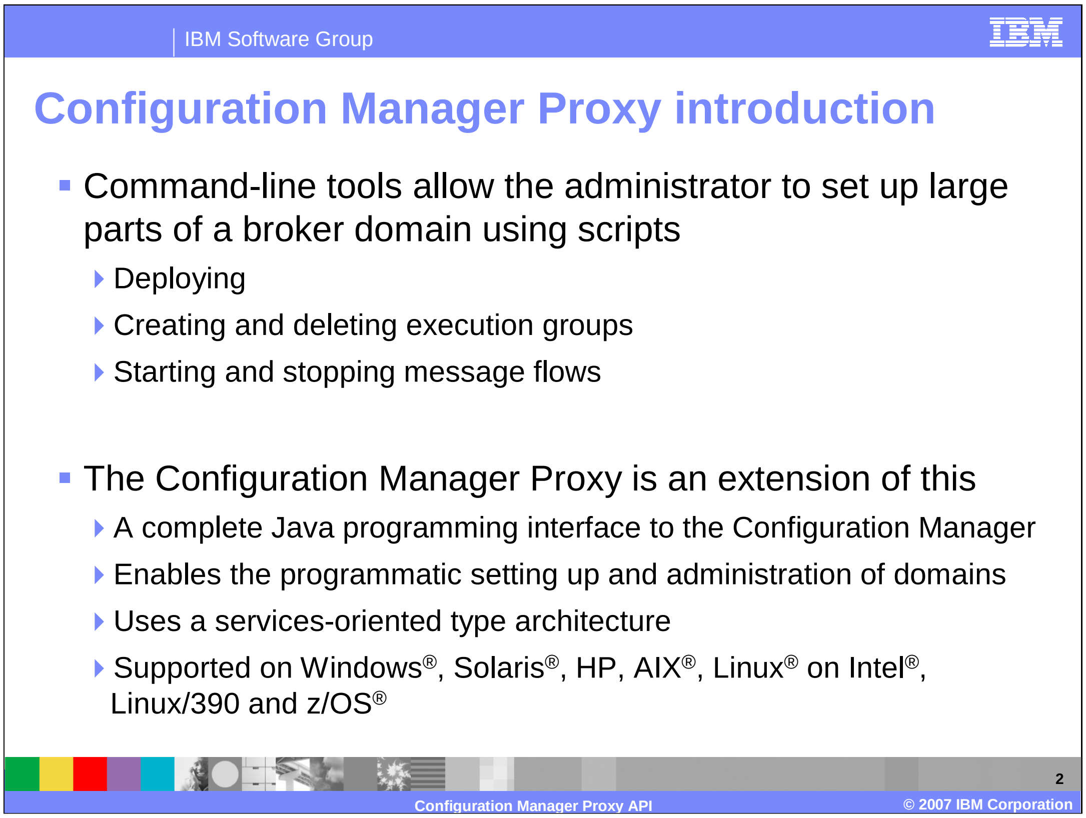
WebSphere® Message Broker Version 6

Configuration Manager Proxy API

business on demand.

Updated March 5, 2007

This presentation discusses the new Configuration Manager Proxy API available in WebSphere Message Broker Version 6.



IBM Software Group

Configuration Manager Proxy introduction

- Command-line tools allow the administrator to set up large parts of a broker domain using scripts
  - Deploying
  - Creating and deleting execution groups
  - Starting and stopping message flows
- The Configuration Manager Proxy is an extension of this
  - A complete Java programming interface to the Configuration Manager
  - Enables the programmatic setting up and administration of domains
  - Uses a services-oriented type architecture
  - Supported on Windows®, Solaris®, HP, AIX®, Linux® on Intel®, Linux/390 and z/OS®

2

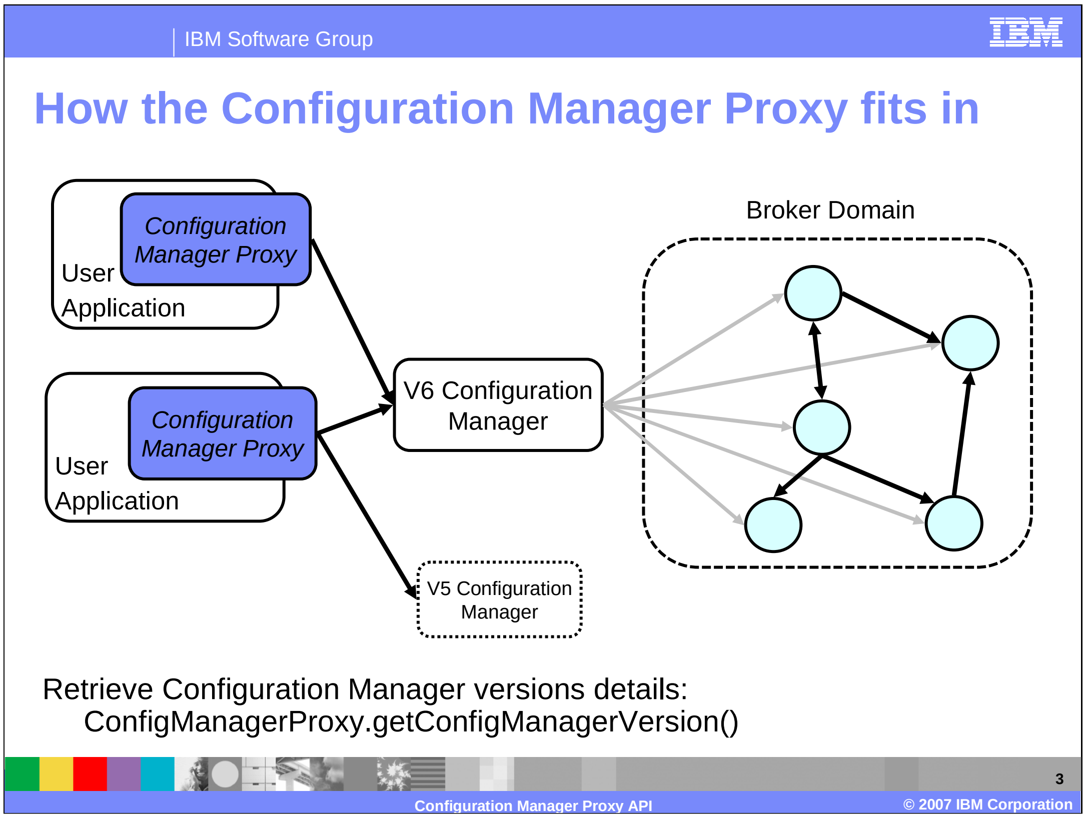
Configuration Manager Proxy API

© 2007 IBM Corporation

The Configuration Manager Proxy API is a new function in WebSphere Message Broker Version 6 which provides a programmatic way to create and update resources required in a broker environment.

This adds to the existing techniques provided by the broker Toolkit and the broker Command Line tools. These tools themselves use the Proxy API to achieve the requested function, and this interface has now been exposed externally as a fully supported component of WebSphere Message Broker Version 6.

The Proxy API is a Java programming interface, and allows an application to directly read and update resources in the Configuration Manager. All functions are provided through this interface, and it is available on all platforms supported by the broker.



IBM Software Group

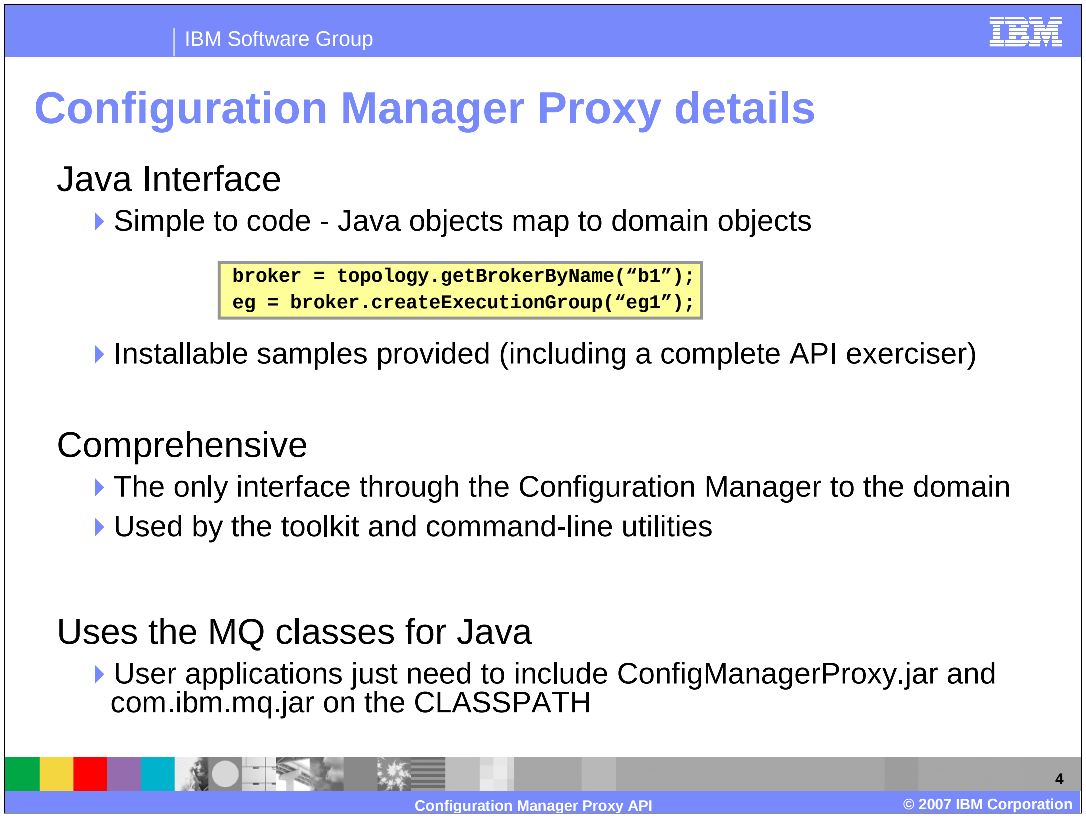
How the Configuration Manager Proxy fits in

Retrieve Configuration Manager versions details:
ConfigManagerProxy.getConfigManagerVersion()

This slide shows how the Proxy API fits in to user applications. Each user application has access to an imbedded jar file, which contains the Proxy API. These will be located in the class path used by the user application. The jar file is called “ConfigManagerProxy.jar”. The methods contained within this jar file access the attached Configuration Manager, which is in turn used to manage the broker.

As shown on this slide, the user application can control both a Version 6 Configuration Manager and a Version 5 Configuration Manager.

The user application can discover which version of Configuration Manager is connected by issuing the command shown at bottom of the slide. The response will be either 5 or 6, and this will determine which precise functions are available to the calling application. The user application cannot connect to versions earlier than 5.



Configuration Manager Proxy details

Java Interface

- Simple to code - Java objects map to domain objects

```
broker = topology.getBrokerByName(“b1”);
eg = broker.createExecutionGroup(“eg1”);
```

- Installable samples provided (including a complete API exerciser)

Comprehensive

- The only interface through the Configuration Manager to the domain
- Used by the toolkit and command-line utilities

Uses the MQ classes for Java

- User applications just need to include ConfigManagerProxy.jar and com.ibm.mq.jar on the CLASSPATH

This slide shows the details of how to invoke the Proxy API. The implementation of this API has been done by mapping Java objects onto objects in the broker domain.

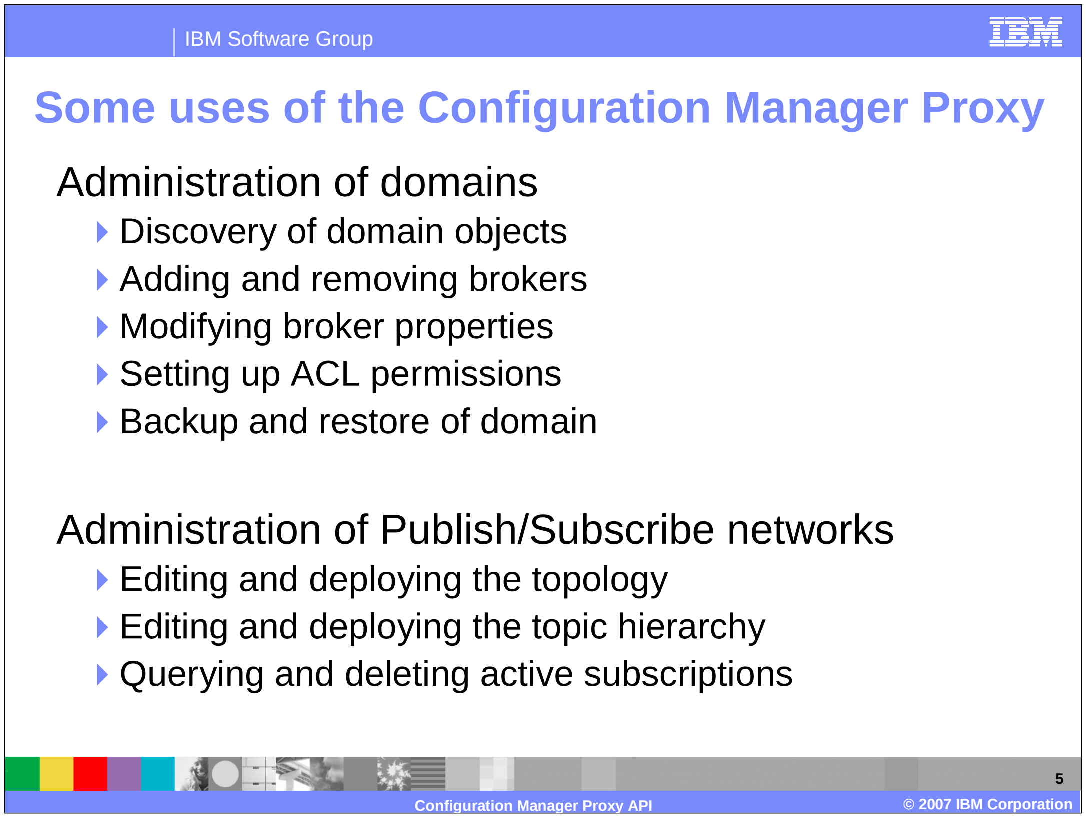
Shown in the yellow box is an object called “broker”. The application wants to obtain the topology of the broker called “b1” by calling the method “getBrokerByName”.

The second line in the example sets the “eg” variable, by invoking the method “createExecutionGroup” on the variable “broker”, specifying the value “eg1” for the name of the execution group. Further examples are provided through the Samples Gallery in the broker Toolkit.

The Proxy API is also used by the broker Toolkit Administration perspective, and the broker Line Commands, such as *mqsideploy*. This ensures consistency across all applications that access the Configuration Manager, regardless of whether they are user-written or part of the broker product.

Access to the Proxy API from a user application is still subject to security control using Access Control Lists. Hence, if you are not permitted access to the Configuration Manager through the broker Toolkit, then the same restriction will apply if you access the same resource through the Proxy API.

The Proxy API classes connect to the Configuration Manager through the WebSphere MQ Java Client and the associated queue manager. Hence, if you are using this function on a system which has no broker or Toolkit installation, it will be necessary to make the MQ Java classes available to the user application.



IBM Software Group

Some uses of the Configuration Manager Proxy

Administration of domains

- Discovery of domain objects
- Adding and removing brokers
- Modifying broker properties
- Setting up ACL permissions
- Backup and restore of domain

Administration of Publish/Subscribe networks

- Editing and deploying the topology
- Editing and deploying the topic hierarchy
- Querying and deleting active subscriptions

5

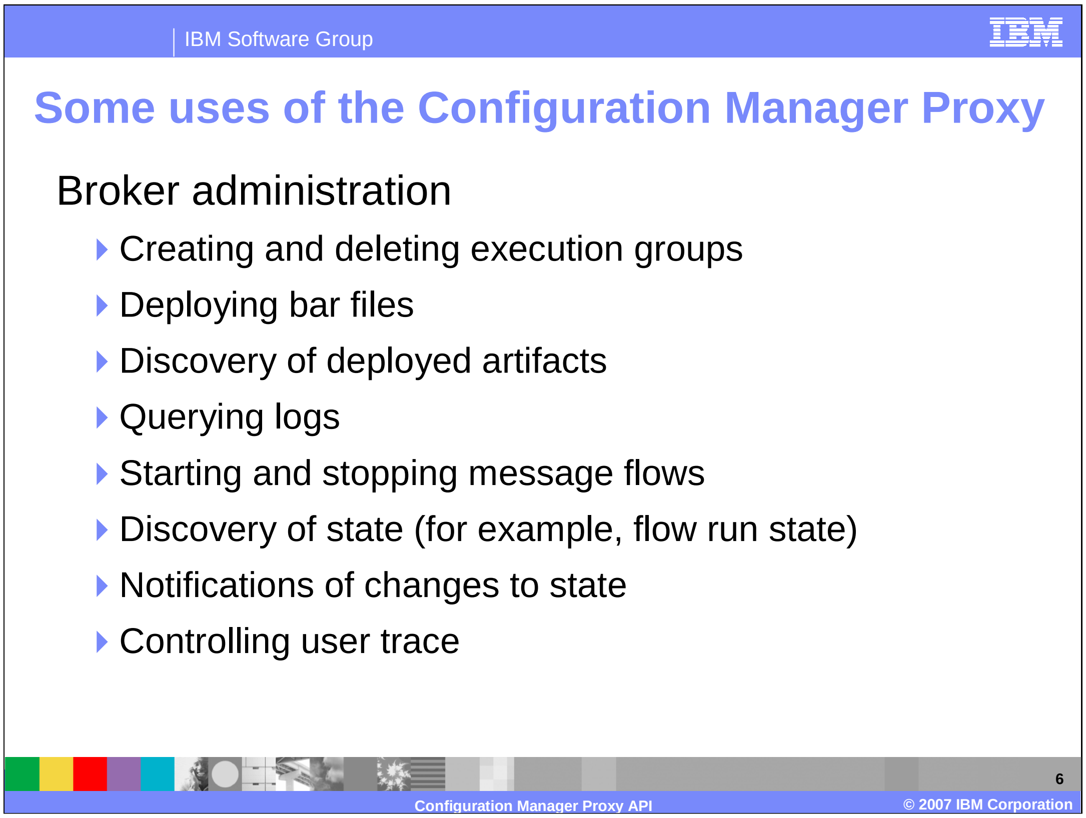
Configuration Manager Proxy API

© 2007 IBM Corporation

The Configuration Manager Proxy API can be used to discover existing objects, to retrieve their properties, and to add new objects or change properties of existing ones.

It can be used to make changes to the security definitions, based on the ACL mechanisms of the Configuration Manager.

And it can be used to manage the publish/subscribe topology.



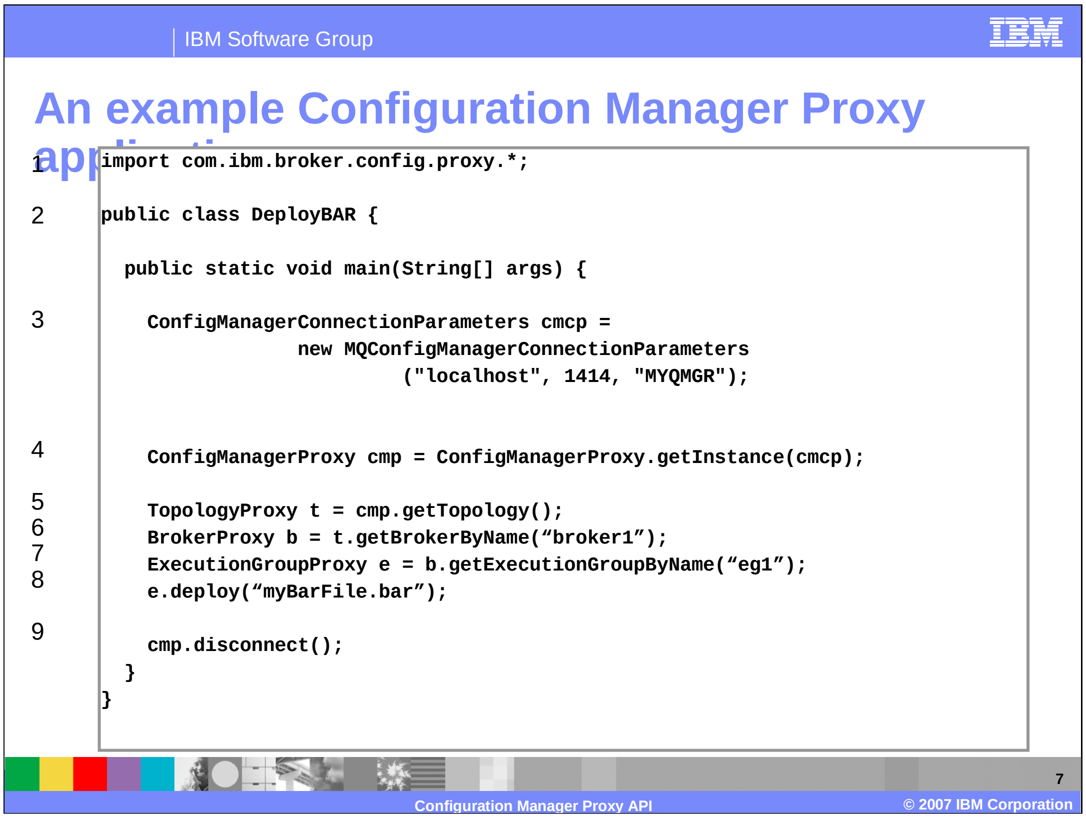
Some uses of the Configuration Manager Proxy

Broker administration

- Creating and deleting execution groups
- Deploying bar files
- Discovery of deployed artifacts
- Querying logs
- Starting and stopping message flows
- Discovery of state (for example, flow run state)
- Notifications of changes to state
- Controlling user trace

This slide shows further examples of possible functions available through the Proxy API.

The Proxy API can be invoked by a Java application that is itself invoked by a JavaCompute node in a broker message flow. Hence, access to the API can be invoked on the basis of some event that happens within a Message Flow. For example, a Message Flow may detect some particular situation indicated by a certain data pattern. This situation may then result in the Message Flow invoking a certain path, which invokes a particular function through the Proxy API, such as “Stop Message Flow”.



An example Configuration Manager Proxy application

```
1   import com.ibm.broker.config.proxy.*;

2   public class DeployBAR {

      public static void main(String[] args) {

3       ConfigManagerConnectionParameters cmcp =
                     new MQConfigManagerConnectionParameters
                              ("localhost", 1414, "MYQMGR");


4       ConfigManagerProxy cmp = ConfigManagerProxy.getInstance(cmcp);

5       TopologyProxy t = cmp.getTopology();
6       BrokerProxy b = t.getBrokerByName(“broker1”);
7       ExecutionGroupProxy e = b.getExecutionGroupByName(“eg1”);
8       e.deploy(“myBarFile.bar”);

9       cmp.disconnect();
      }
    }
```

This slide shows an example of a Java application invoking the Proxy API.

The import statement on Line 1 says “use the broker Configuration Manager API classes”. All the required classes are contained in the package shown on this line.

Line 2 indicates that this application is going to deploy a bar file.

Line 3 sets up the connection parameters that will be used to connect to the Configuration Manager.

Line 4 connects to the Configuration Manager, based on the parameters in Line 3. There is now an object called “cmp”, which represents the connection to the active Configuration Manager.

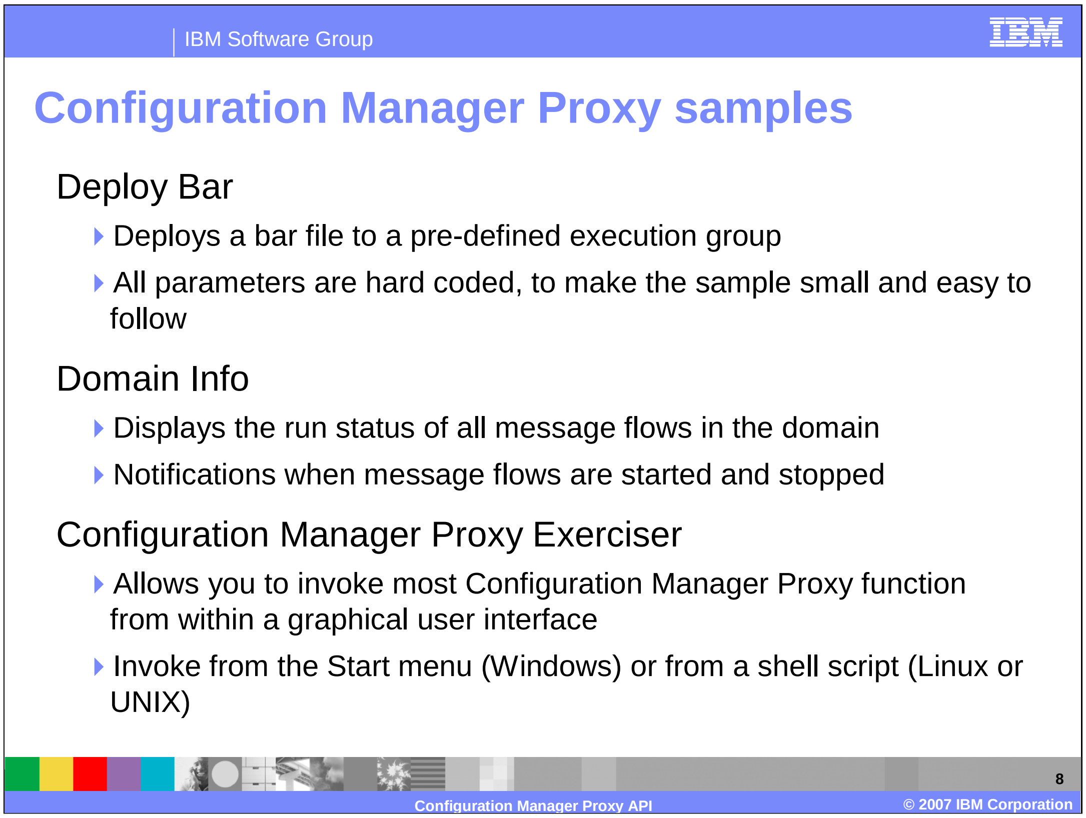
Lines 5 and 6 then retrieve the broker topology, and the specific details about a broker named “broker1”. In this case, the application has hard-coded the name of the required broker, rather than obtaining it from the list of returned broker in the topology. This is represented by the broker object called “b”.

Line 7 then obtains information about the execution group called “eg1” from this broker. This is then represented by the object “e”.

Line 8 then deploys a bar file to the execution group, by referencing the object “e”.

Finally, for good programming practice, Line 9 disconnects from the Configuration Manager.

This example does not include any error checking or handling. For example, if the connection to the Configuration Manager was lost, an error would be thrown when the application attempts to use that connection.



Configuration Manager Proxy samples

Deploy Bar

- Deploys a bar file to a pre-defined execution group
- All parameters are hard coded, to make the sample small and easy to follow

Domain Info

- Displays the run status of all message flows in the domain
- Notifications when message flows are started and stopped

Configuration Manager Proxy Exerciser

- Allows you to invoke most Configuration Manager Proxy function from within a graphical user interface
- Invoke from the Start menu (Windows) or from a shell script (Linux or UNIX)

WebSphere Message Broker Version 6 ships three samples for the Proxy API. These samples can be found at *INST_DIR*/sample/ConfigManagerProxy where *INST_DIR* is the installation directory. The CMP samples can be run as they are shipped, or they can be modified. Both the executables and source files (in folder cmp) are available.

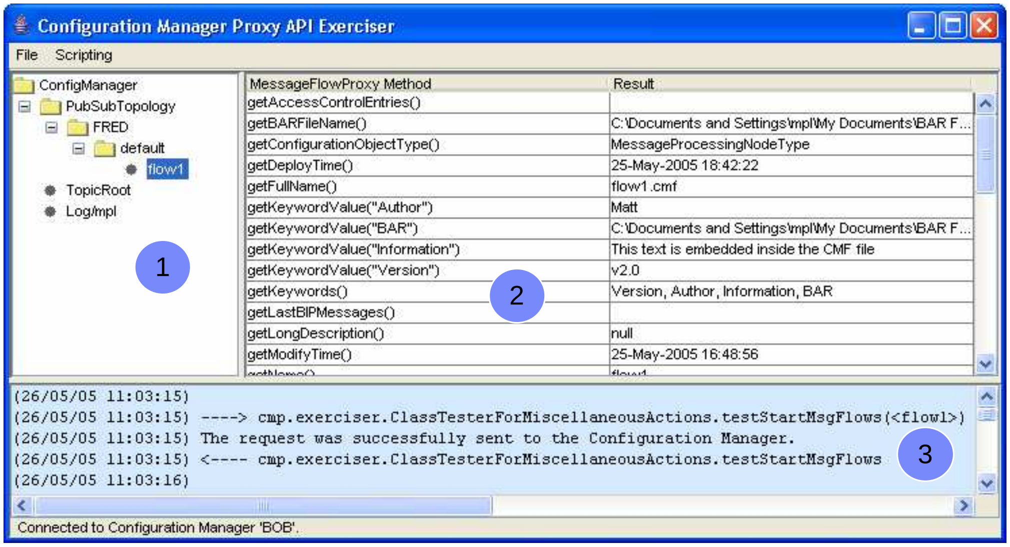
The Deploy Bar sample was shown on the previous slide.

The Domain Info sample examines all objects deployed into the broker, across all execution groups. It reports on the status of these, and reports whether they are running or stopped. This sample also includes a facility to provide an alert whenever any message flows are started or stopped.

The Configuration Manager Proxy Exerciser is a full-function java application with a GUI interface, providing access to the complete API through a graphical interface. This is provided on the Windows, Linux and UNIX platforms.

IBM Software Group

Configuration Manager Proxy API Exerciser



MessageFlowProxy Method	Result
getAccessControlEntries()	
getBARFileName()	C:\Documents and Settings\mpl\My Documents\BAR F...
getConfigurationObjectType()	MessageProcessingNodeType
getDeployTime()	25-May-2005 18:42:22
getFullName()	flow1.cmf
getKeywordValue("Author")	Matt
getKeywordValue("BAR")	C:\Documents and Settings\mpl\My Documents\BAR F...
getKeywordValue("Information")	This text is embedded inside the CMF file
getKeywordValue("Version")	v2.0
getKeywords()	Version, Author, Information, BAR
getLastBIPMessages()	
getLongDescription()	null
getModifyTime()	25-May-2005 16:48:56
getName()	flow1

```
(26/05/05 11:03:15)
(26/05/05 11:03:15) ----> cmp.exerciser.ClassTesterForMiscellaneousActions.testStartMsgFlows(<flow1>)
(26/05/05 11:03:15) The request was successfully sent to the Configuration Manager.
(26/05/05 11:03:15) <---- cmp.exerciser.ClassTesterForMiscellaneousActions.testStartMsgFlows
(26/05/05 11:03:16)
```

Connected to Configuration Manager 'BOB'.

1. Domain objects (right-click to manipulate)

2. Information on the selected object (API invocations)

3. Output from API calls (pointers to sample source)

9

Configuration Manager Proxy API

© 2007 IBM Corporation

This is a screen print of the Configuration Manager Proxy API Exerciser. This function does not need the broker Toolkit to be installed, but it does require the ability to connect to the Configuration Manager.

To start the CMP Exerciser on Windows, click Start > IBM WebSphere Message Brokers 6.0 > Java Programming APIs > Configuration Manager Proxy API Exerciser. On other platforms, run the following shell script from the installation directory:*INST_DIR*\sample\ConfigManagerProxy\StartConfigManagerProxyExerciser where *INST_DIR* is the installation directory.

Alternatively, in a Windows environment, open the WebSphere Message Broker product from the Start Menu. One of the options shown is “Java Programming APIs”. This in turn has three options, namely:

•Configuration Manager Proxy API documentation

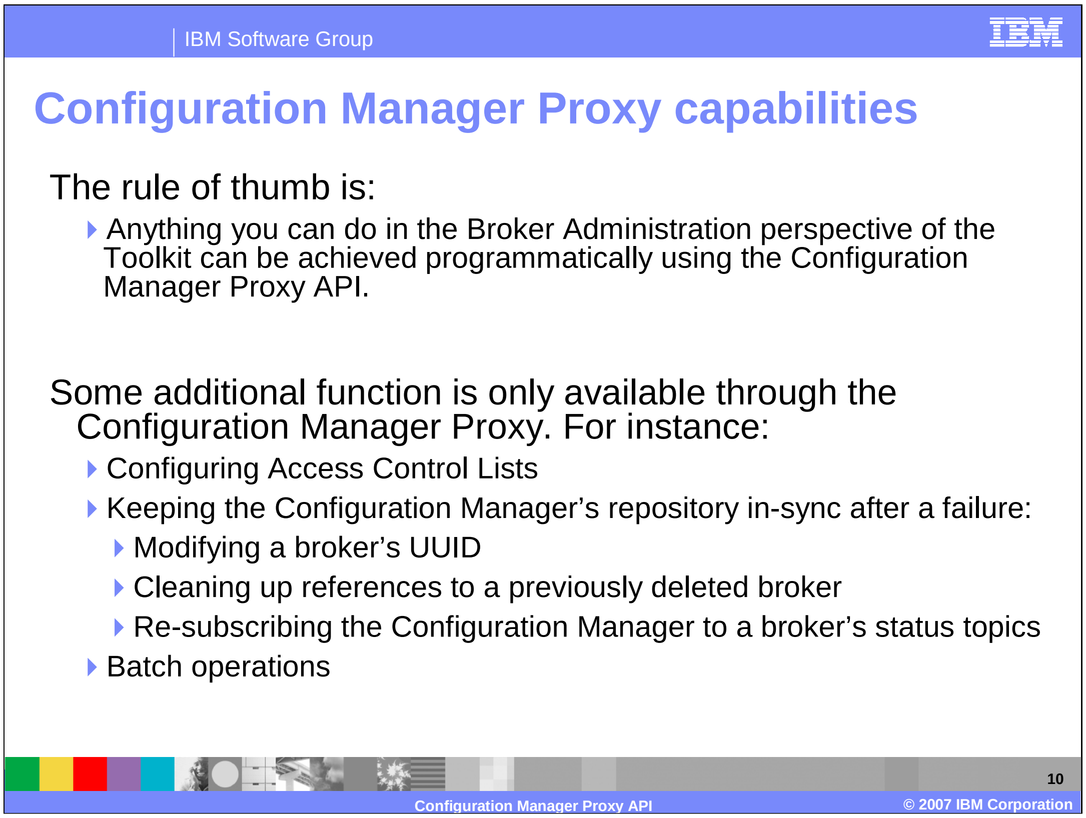
•Configuration Manager Proxy Exerciser

•Java Plug-in API Documentation

When started, the Exerciser shows a blank display. Connect to the required Configuration Manager by right-clicking the Configuration Manager name in the top left-hand corner, and then click Connect. This opens a windows where you provide the connection details for the required Configuration Manager, such as Queue Manager name, TCP/IP address, and port number. Alternatively, you can use the connection file that is generated by the broker Toolkit.

Once connected, the Exerciser will display information about the selected broker objects. Left-clicking an object will retrieve information about the object, and display this information in the main panel. As an example, the slide shows that the object “flow1” has been clicked, shown in Area 1. Information about the message flow “flow1” is displayed in Area 2.

To perform an action on the broker objects, use the mouse right-click on the object to display the permissible actions for that object. Clicking on the required action will pass the command to the Configuration Manager, and the result of the action will be shown in the log, in Area 3. For example, if the broker is highlighted, the Exerciser can then create a new execution group in the broker, by right-clicking on the broker, and then specifying the name of the new execution group.



IBM Software Group

Configuration Manager Proxy capabilities

The rule of thumb is:

- Anything you can do in the Broker Administration perspective of the Toolkit can be achieved programmatically using the Configuration Manager Proxy API.

Some additional function is only available through the Configuration Manager Proxy. For instance:

- Configuring Access Control Lists
- Keeping the Configuration Manager's repository in-sync after a failure:
  - Modifying a broker's UUID
  - Cleaning up references to a previously deleted broker
  - Re-subscribing the Configuration Manager to a broker's status topics
- Batch operations

10

Configuration Manager Proxy API

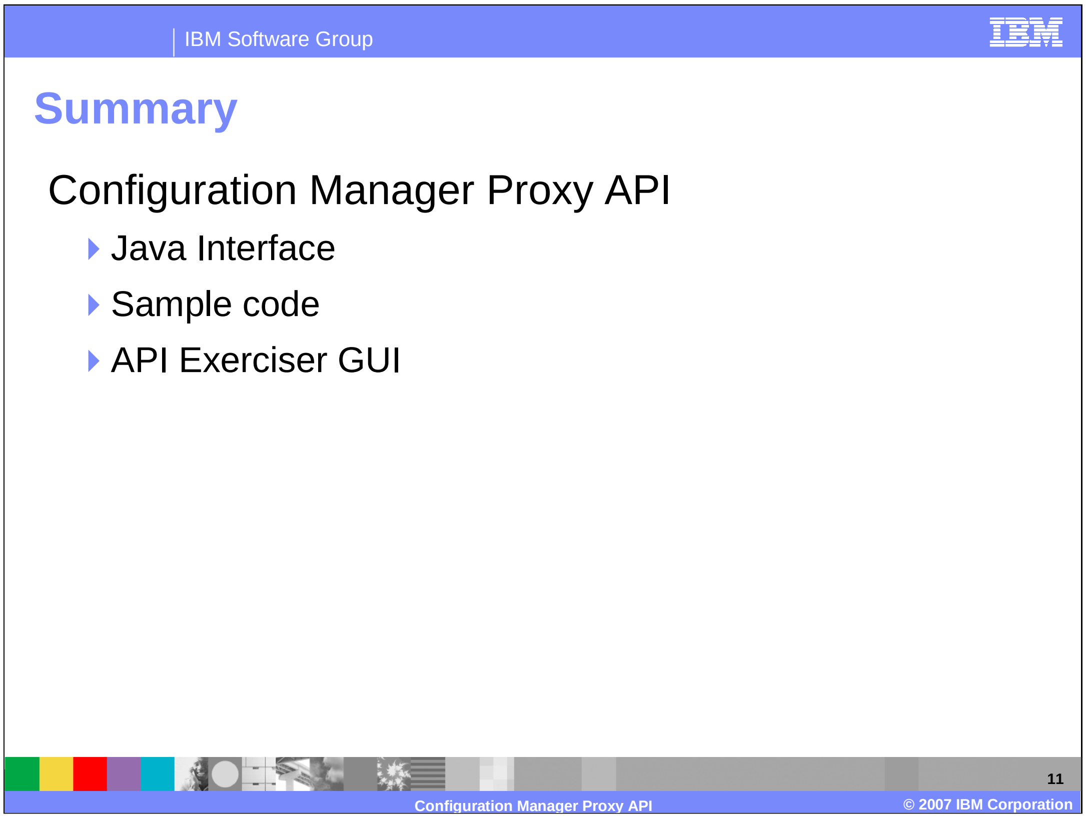
© 2007 IBM Corporation

The Configuration Manager Proxy API provides all of the function that is available through the Message Broker Toolkit.

In addition, there are some extra functions available through the API that are not available through the Toolkit.

The Proxy API, and the API Exerciser, can be used to create and maintain Configuration Manager Access Control Lists, or ACLs. The user with necessary authorities can update the ACL database within the Configuration Manager.

If a failure occurs with the Configuration Manager, resulting in the database being out of sync with the broker, the Proxy API provides a facility to correct this.

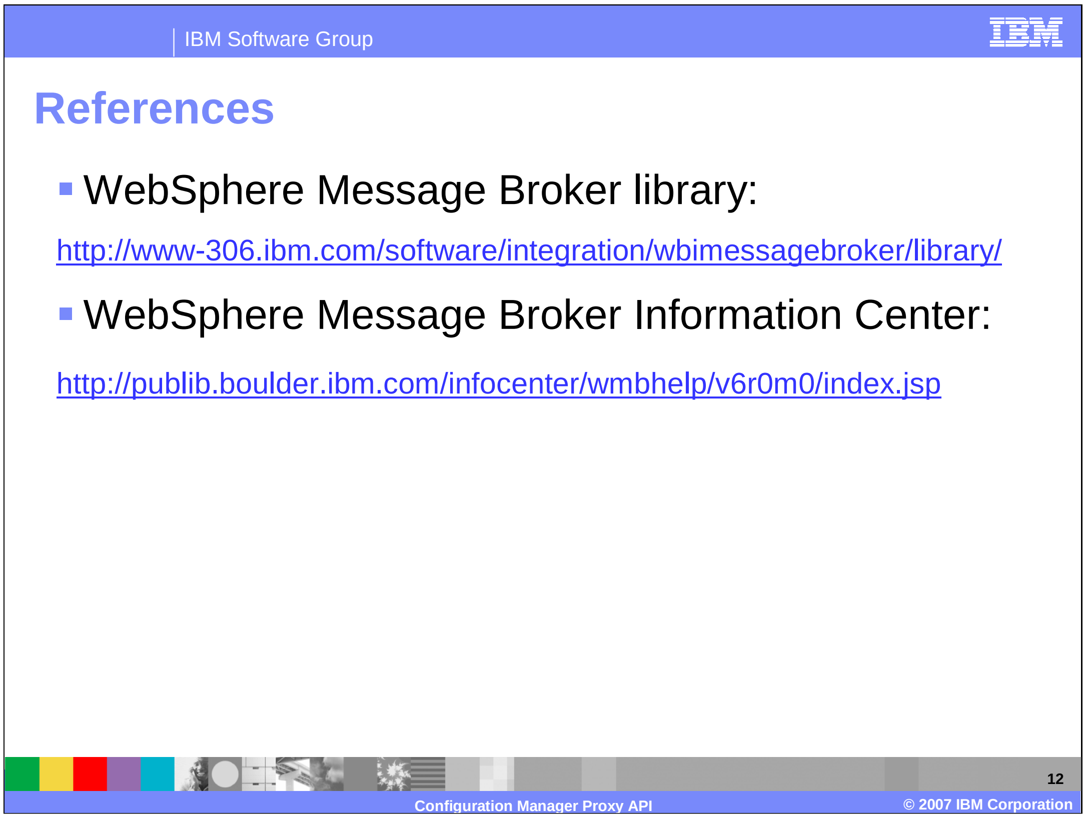


Summary

Configuration Manager Proxy API

- Java Interface
- Sample code
- API Exerciser GUI

This session covered the Java API provided for the Configuration Manager. This is an API provided for use with user applications written in Java and a sample application with a graphical user interface.



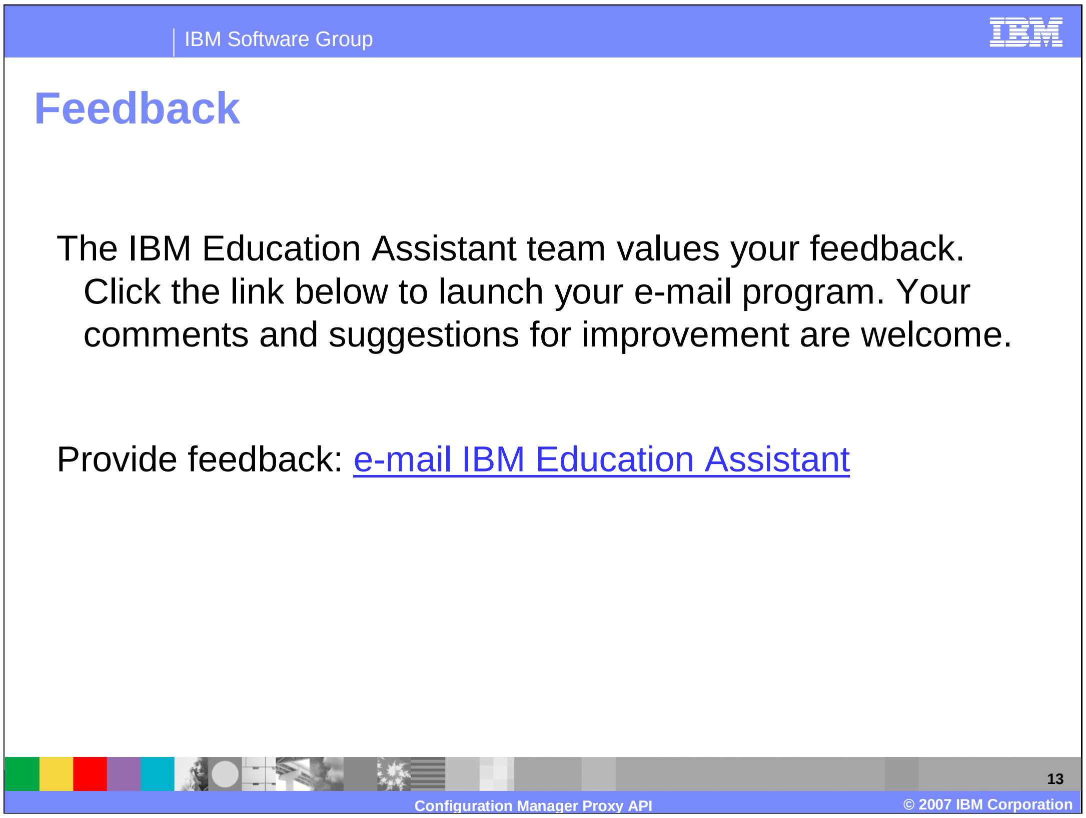
References

- WebSphere Message Broker library:

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

- WebSphere Message Broker Information Center:

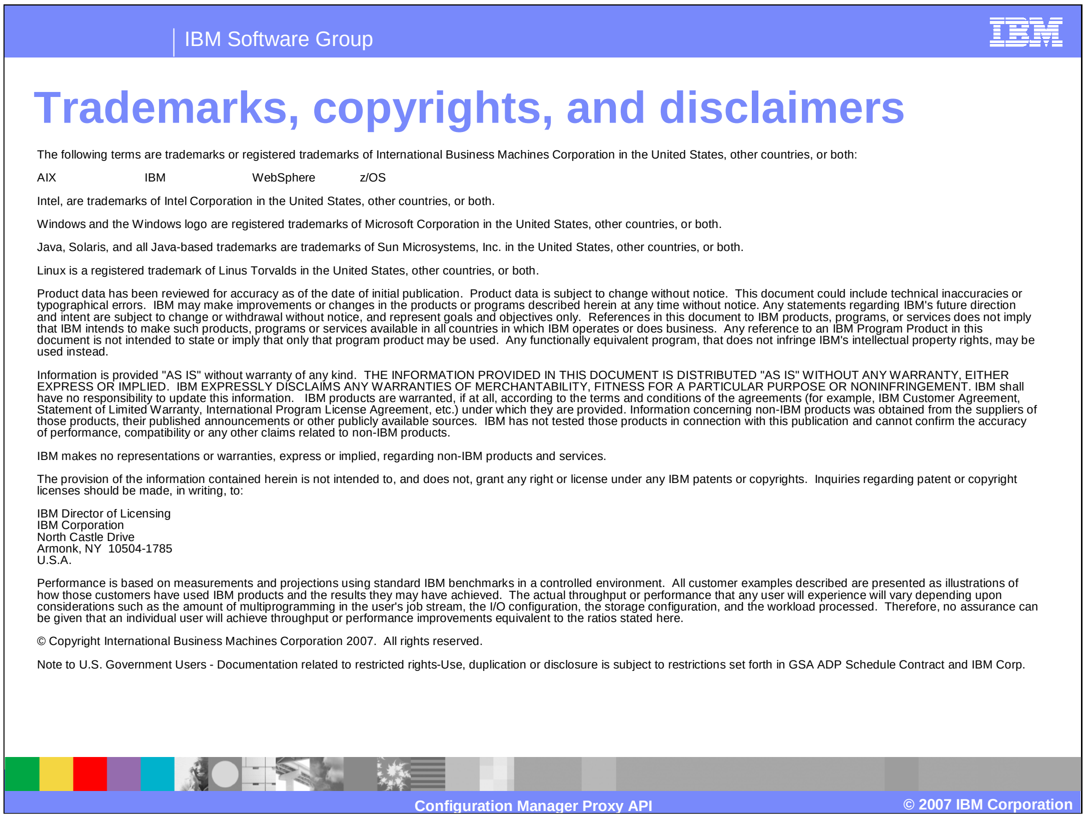
http://publib.boulder.ibm.com/infocenter/wmbhelp/v6r0m0/index.jsp



Feedback

The IBM Education Assistant team values your feedback. Click the link below to launch your e-mail program. Your comments and suggestions for improvement are welcome.

Provide feedback: e-mail IBM Education Assistant



Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

AIX IBM WebSphere z/OS

Intel, are trademarks of Intel Corporation in the United States, other countries, or both.

Windows and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, Solaris, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.