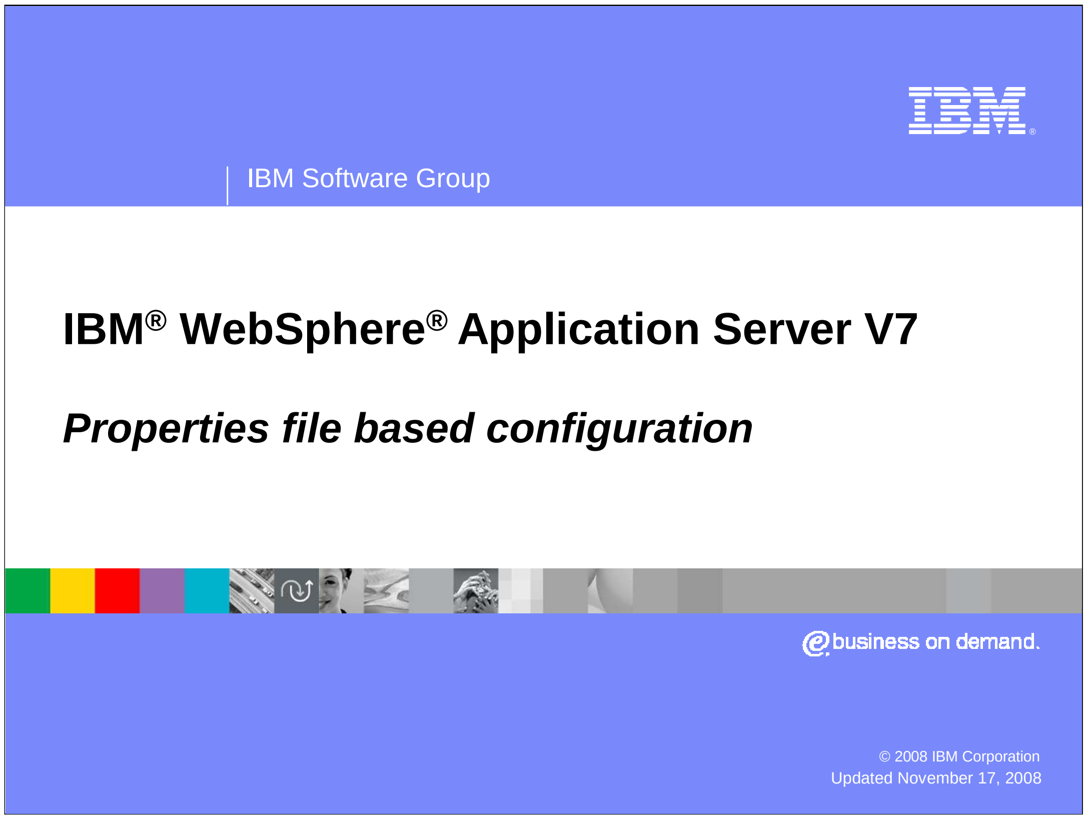


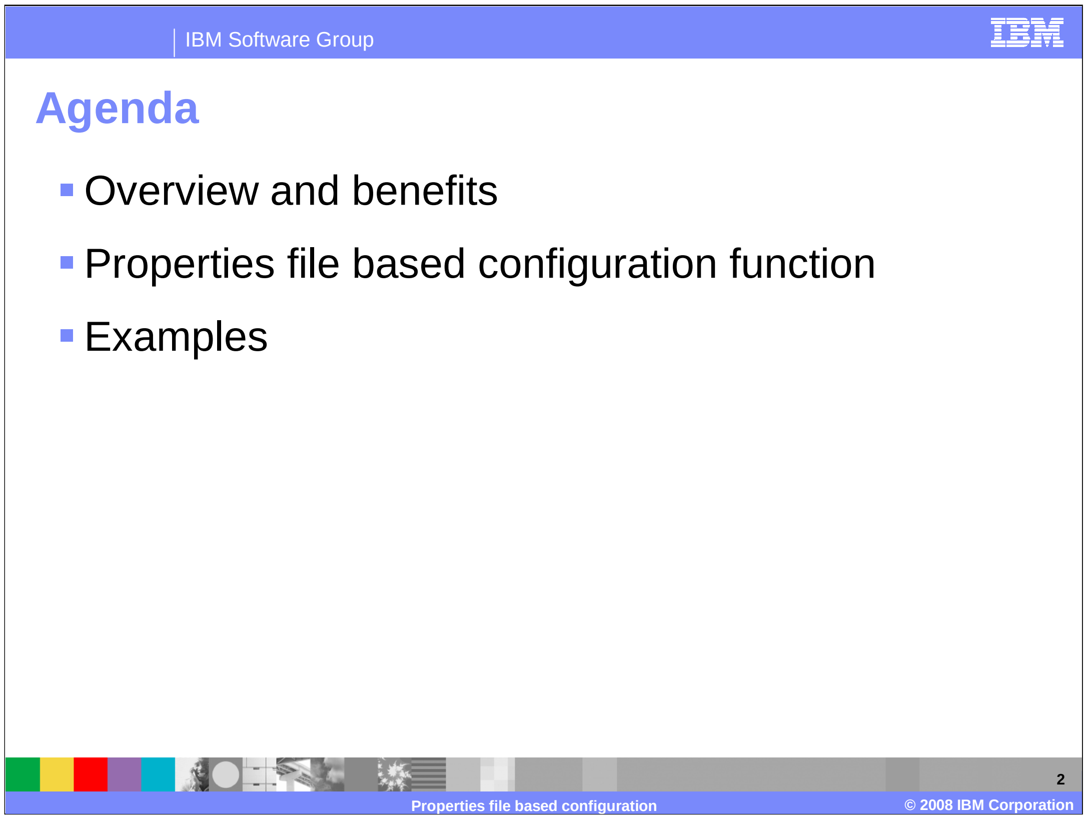
IBM Software Group

IBM® WebSphere® Application Server V7

Properties file based configuration

@business on demand.

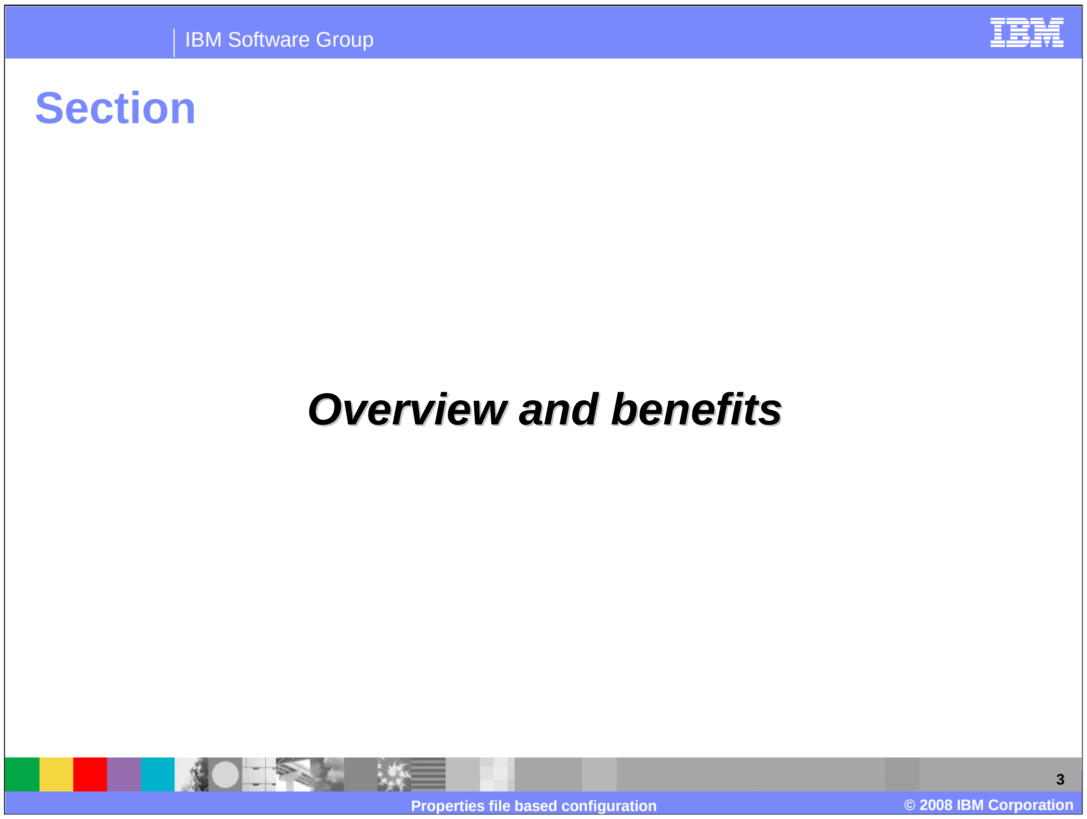
This presentation explains properties file based configuration – a new feature introduced in WebSphere Application Server V7. This feature allows you to convert the multiple XML-formatted and other complex configuration files that make up the configuration repository into a simple properties file format that contains the same configuration information.



Agenda

- Overview and benefits
- Properties file based configuration function
- Examples

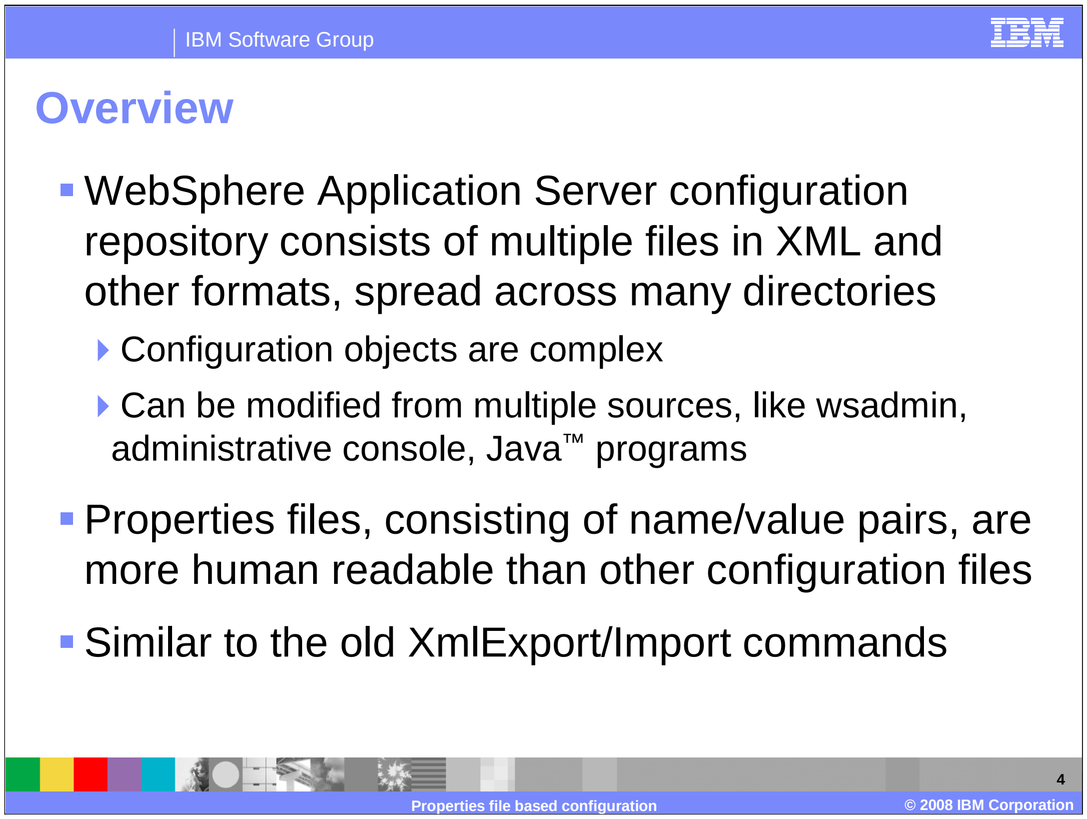
The first section provides an overview of properties file based configuration, and the second section describes the functionality that is available with the new properties file configuration commands. The last section includes several examples of how to use the new properties file commands to work with your application server configuration.



Section

Overview and benefits

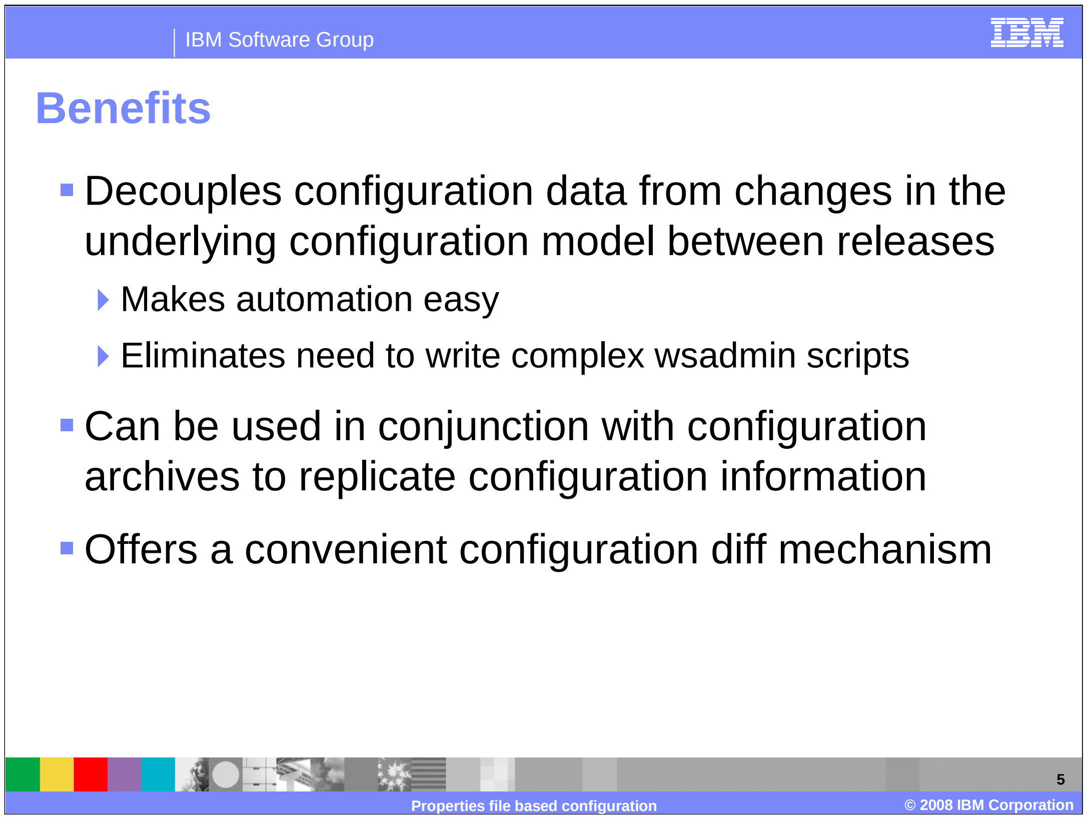
This section contains an overview of properties file based configuration and describes some benefits of this new configuration scheme.



Overview

- WebSphere Application Server configuration repository consists of multiple files in XML and other formats, spread across many directories
  - Configuration objects are complex
  - Can be modified from multiple sources, like wsadmin, administrative console, Java™ programs
- Properties files, consisting of name/value pairs, are more human readable than other configuration files
- Similar to the old XmlExport/Import commands

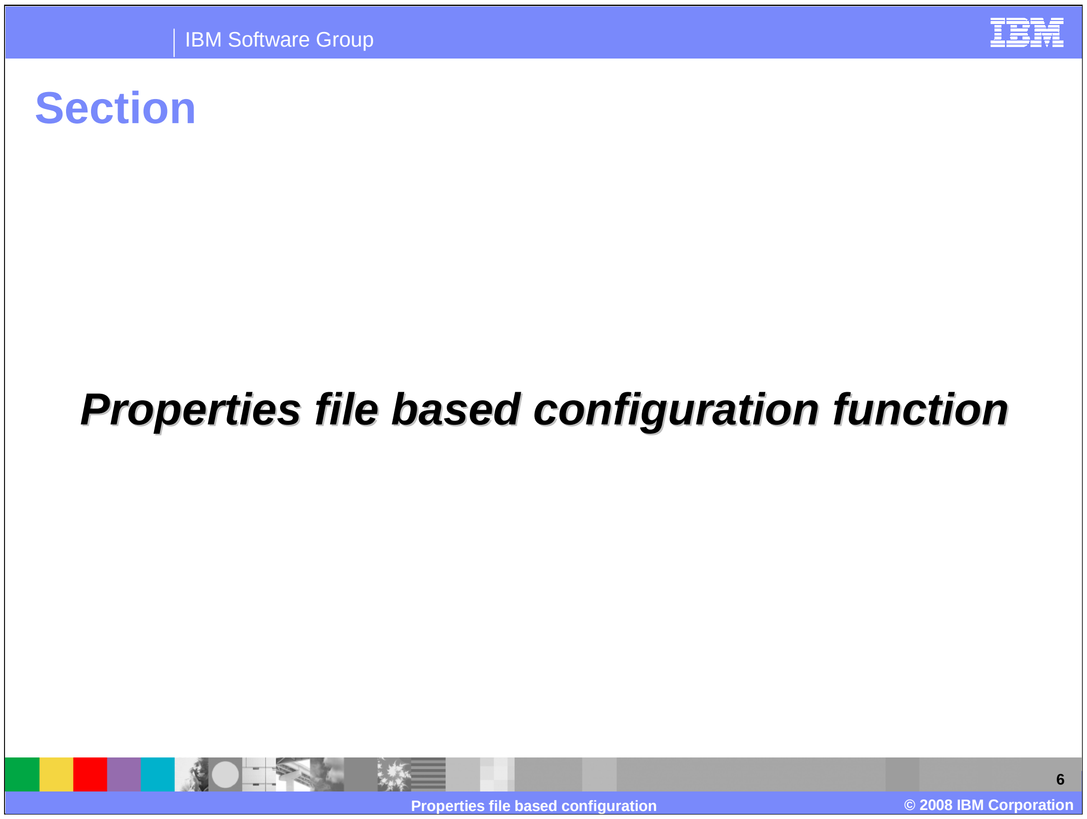
The WebSphere Application Server configuration repository consists of many files in different formats spread across multiple directories. Some files are in XML format, and others contain complex configuration objects associated with the WebSphere Common Configuration Model. In the previous release, users relied on wsadmin, the administrative console, and Java APIs to query and modify these configuration objects. Properties file configuration is a new scheme for working with these configuration objects that is based on simple properties files in a standard name/value pair format. A new set of wsadmin commands is available in V7 that can extract and apply properties files to configuration objects. Applying a properties file to your configuration automatically modifies the corresponding objects in the configuration repository, and the properties files are more human readable than many of the files that are kept in the repository. WebSphere derives configuration information from the configuration repository, not from configuration properties files. To update the configuration repository so that it reflects the information in a configuration properties file, you must use wsadmin commands to apply the properties file to the configuration. Some similar commands for exporting and importing XML-formatted configuration files were available in older versions of the application server, but they have not been available for several releases.



Benefits

- Decouples configuration data from changes in the underlying configuration model between releases
  - Makes automation easy
  - Eliminates need to write complex wsadmin scripts
- Can be used in conjunction with configuration archives to replicate configuration information
- Offers a convenient configuration diff mechanism

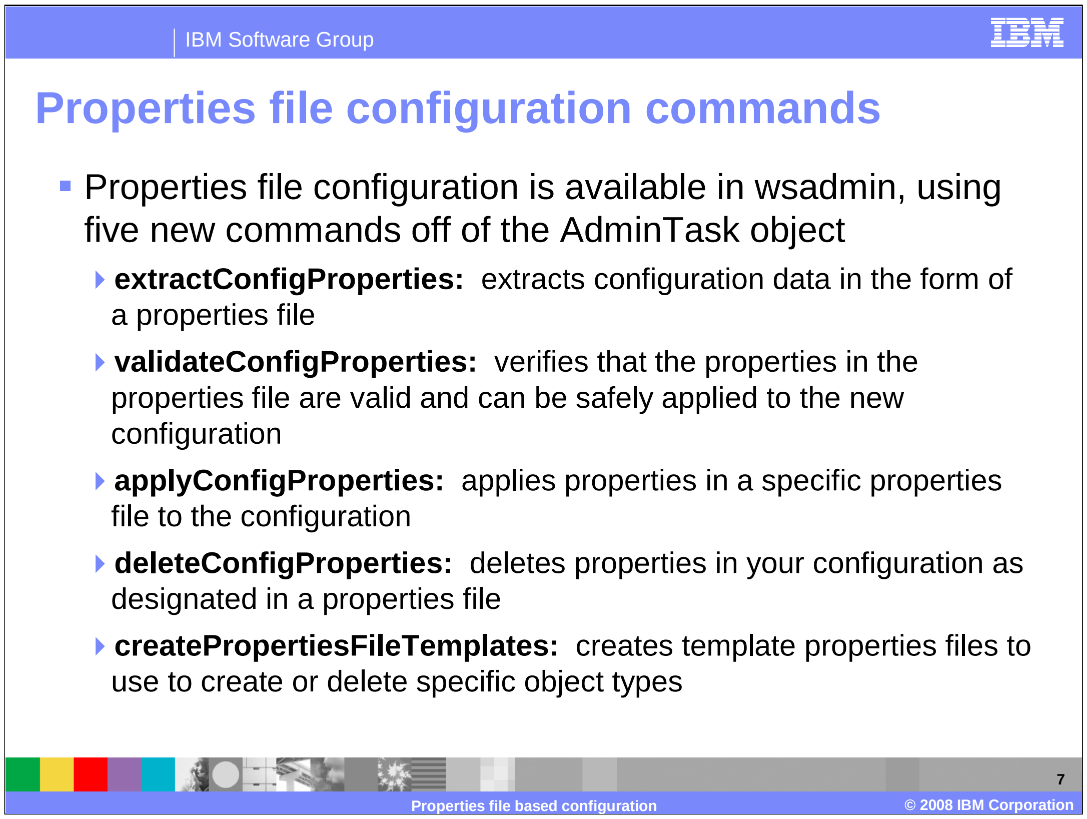
Many users of the application server have tried to create their own automated methods for modifying an application server configuration, often by directly manipulating files in the configuration repository. However, custom solutions are tightly coupled to underlying file structure, which can change from release to release. The new properties file configuration scheme decouples configuration data from changes in the underlying configuration model between releases. This makes it easy to automate the configuration process without having to write complex wsadmin scripts. Properties files can be used in conjunction with configuration archives to replicate a configuration by first importing a configuration archive and then using properties file configuration to customize the environment. Since properties files are simple text files, it is easy to use standard text editors to compare properties files from different configuration environments to help identify potential configuration problems.



Section

Properties file based configuration function

This section describes the functionality that is available with properties file based configuration.


IBM Software Group

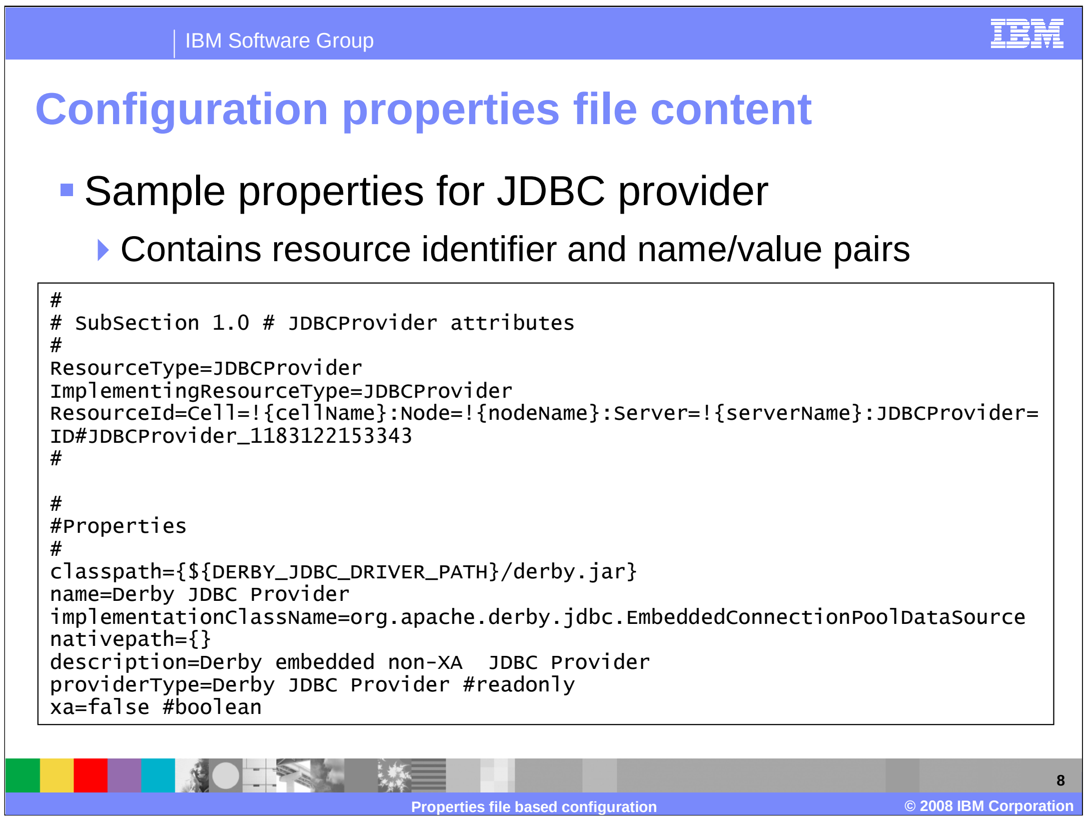
Properties file configuration commands

- Properties file configuration is available in wsadmin, using five new commands off of the AdminTask object
  - **extractConfigProperties:** extracts configuration data in the form of a properties file
  - **validateConfigProperties:** verifies that the properties in the properties file are valid and can be safely applied to the new configuration
  - **applyConfigProperties:** applies properties in a specific properties file to the configuration
  - **deleteConfigProperties:** deletes properties in your configuration as designated in a properties file
  - **createPropertiesFileTemplates:** creates template properties files to use to create or delete specific object types

7

Properties file based configuration © 2008 IBM Corporation

A new PropertiesBasedConfiguration command group has been added to the AdminTask object. This command group contains five commands. The extractConfigProperties command creates a properties file that is based on the current environment. The files can be extracted at different levels, like a cluster, cell, or server, and can be filtered to only contain certain types of configuration attributes, like a JavaVirtualMachine or a data source. Before applying a properties file to your configuration, it is a good practice to validate the format of the file using the validateConfigPropertiesCommand. Once you have validated the file, you can use either the applyConfigProperties or deleteConfigProperties command to process the information in the properties file and modify your configuration. Finally, the createPropertiesFileTemplates command is available to help you create properties file templates for certain actions, like creating a server or installing an application.



IBM Software Group

Configuration properties file content

- Sample properties for JDBC provider
  - Contains resource identifier and name/value pairs

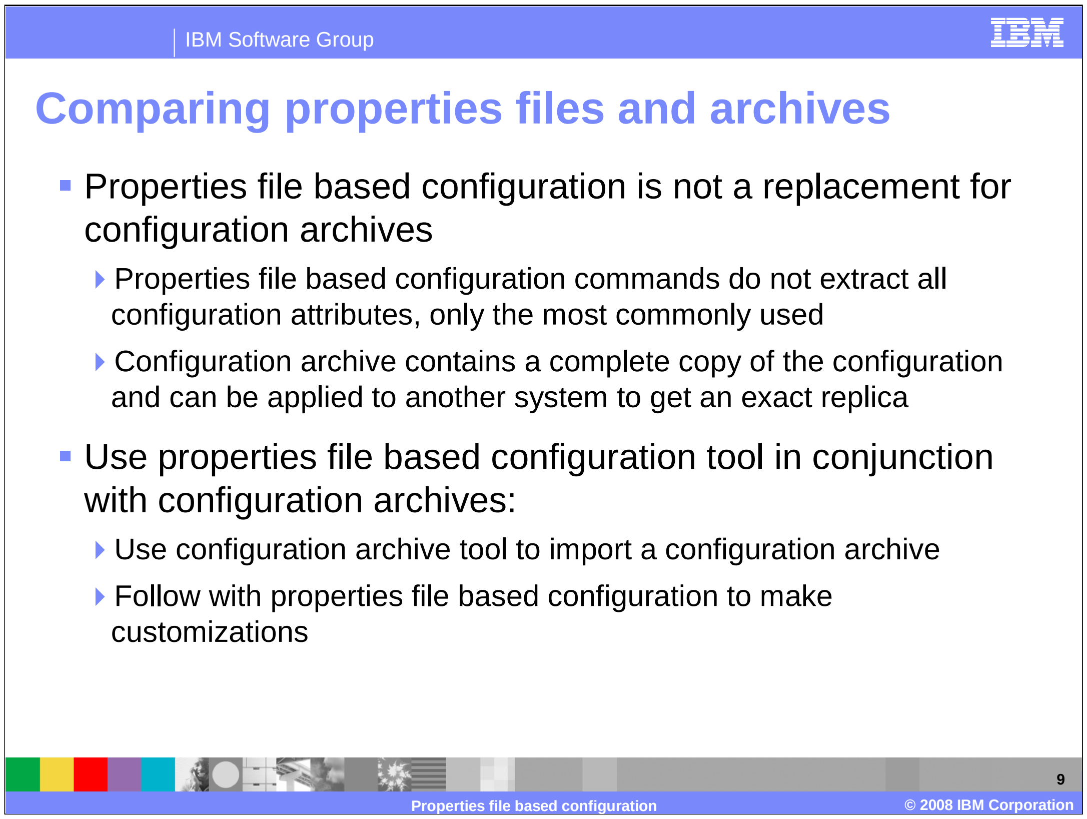
```
#
# SubSection 1.0 # JDBCProvider attributes
#
ResourceType=JDBCProvider
ImplementingResourceType=JDBCProvider
ResourceId=Cell=!{cellName}:Node=!{nodeName}:Server=!{serverName}:JDBCProvider=
ID#JDBCProvider_1183122153343
#

#
#Properties
#
classpath={${DERBY_JDBC_DRIVER_PATH}/derby.jar}
name=Derby JDBC Provider
implementationClassName=org.apache.derby.jdbc.EmbeddedConnectionPoolDataSource
nativepath={}
description=Derby embedded non-XA  JDBC Provider
providerType=Derby JDBC Provider #readonly
xa=false #boolean
```

8

Properties file based configuration © 2008 IBM Corporation

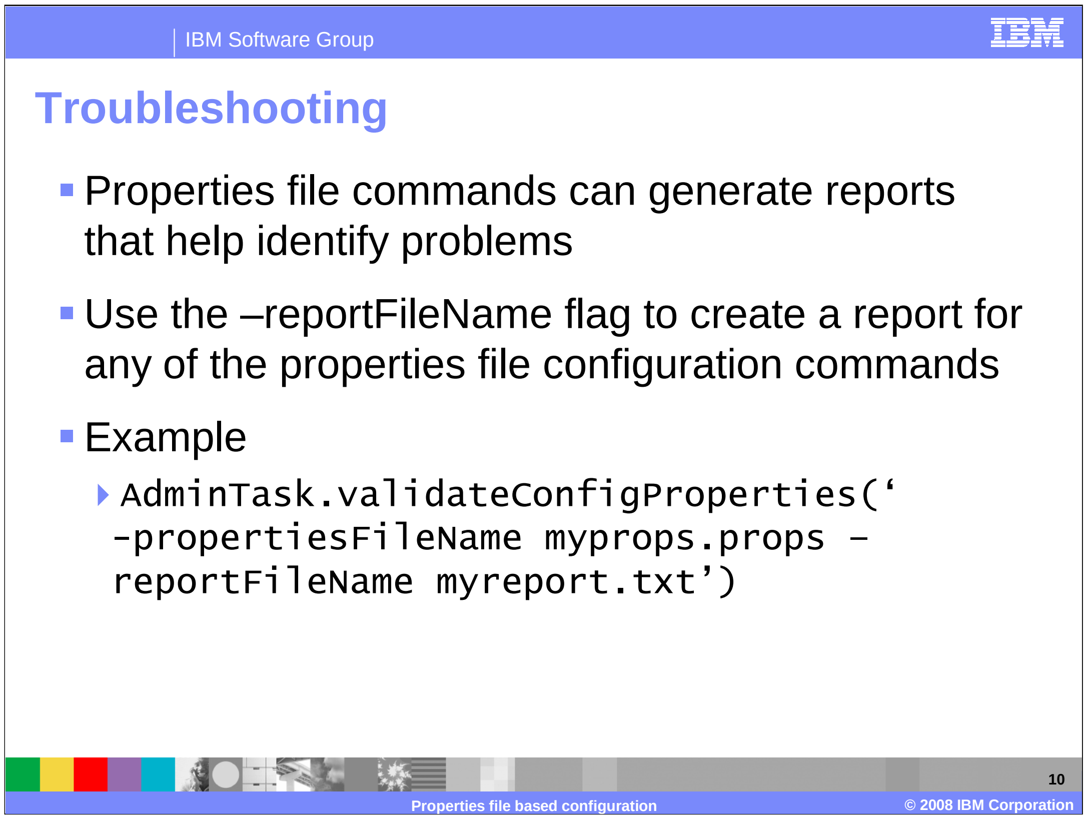
Configuration properties files contain a series of name/value pairs. Each configuration object is defined in a separate section; the example shown here is for a JDBC provider. The first section contains the type of resource – in this case, a JDBC provider – and a resource identifier. The identifier is often in the format shown here, including the cell, node, and server names, and ending with a string that contains the resource type and a large number. The configuration information for this JDBC provider resource is described in the second half of the example, using name/value pairs.



Comparing properties files and archives

- Properties file based configuration is not a replacement for configuration archives
  - Properties file based configuration commands do not extract all configuration attributes, only the most commonly used
  - Configuration archive contains a complete copy of the configuration and can be applied to another system to get an exact replica
- Use properties file based configuration tool in conjunction with configuration archives:
  - Use configuration archive tool to import a configuration archive
  - Follow with properties file based configuration to make customizations

Although properties file based configuration does do some of the things that can be done with WebSphere configuration archives, it is not a replacement. At most, the properties files based configuration commands only extract commonly used configuration attributes from the configuration repository. However, a back up made using configuration archives can contain an exact copy of the configuration that can be applied to another system to exactly replicate the configuration information. If you want to replicate the configuration of one system onto another, with some customizations, you can use configuration archives to gain an exact replica of the configuration and then follow with properties based configuration to make the required customizations.

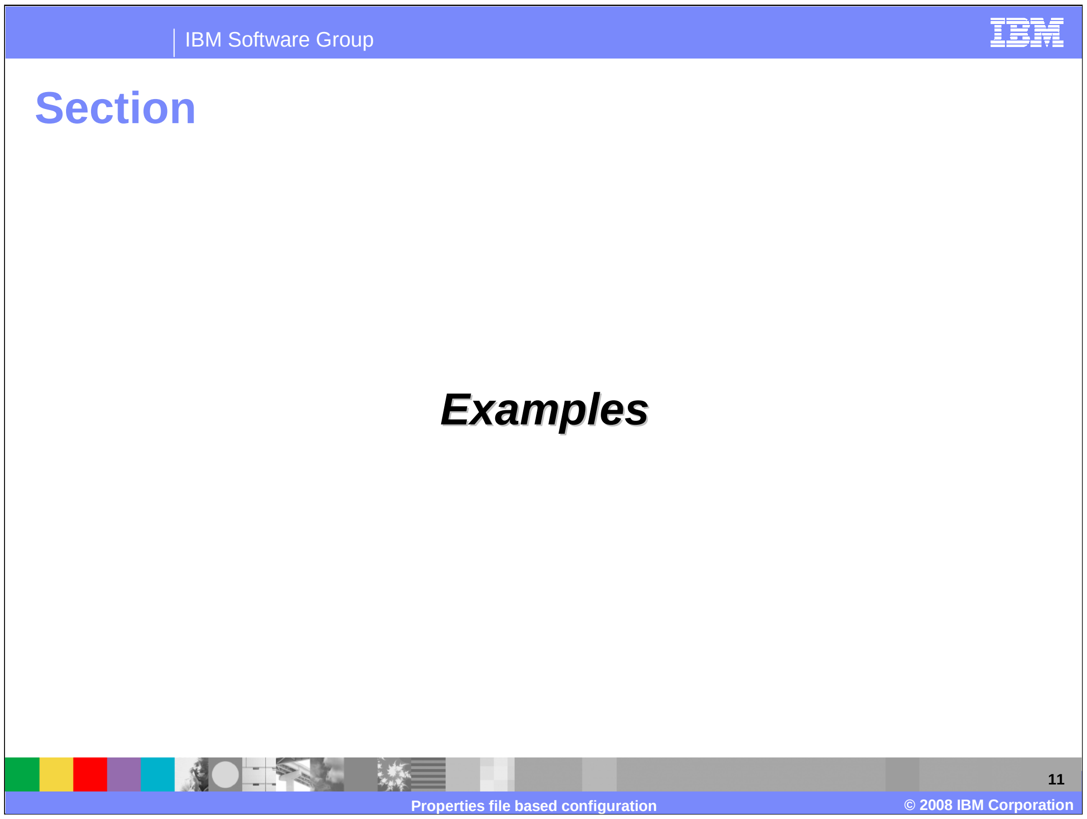


Troubleshooting

- Properties file commands can generate reports that help identify problems
- Use the –reportFileName flag to create a report for any of the properties file configuration commands
- Example

```
AdminTask.validateConfigProperties('
-propertiesFileName myprops.props –
reportFileName myreport.txt')
```

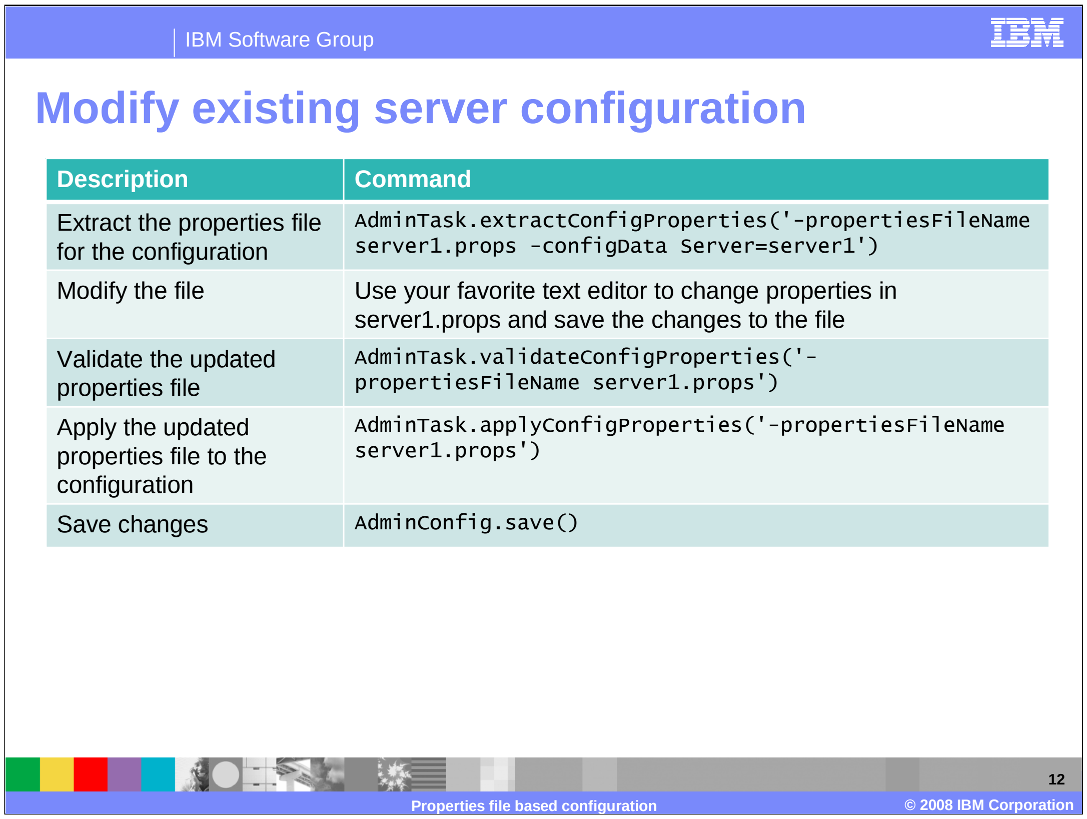
If you encounter any problems while running the properties file commands, you can generate a report file that contains details of the command processing that can help you identify and resolve issues. For example, if the properties file that you are trying to validate contains errors, the report will help you locate and fix the errors in your properties file. The report file is also a helpful resource to provide to IBM Support if you are opening a service request.



Section

Examples

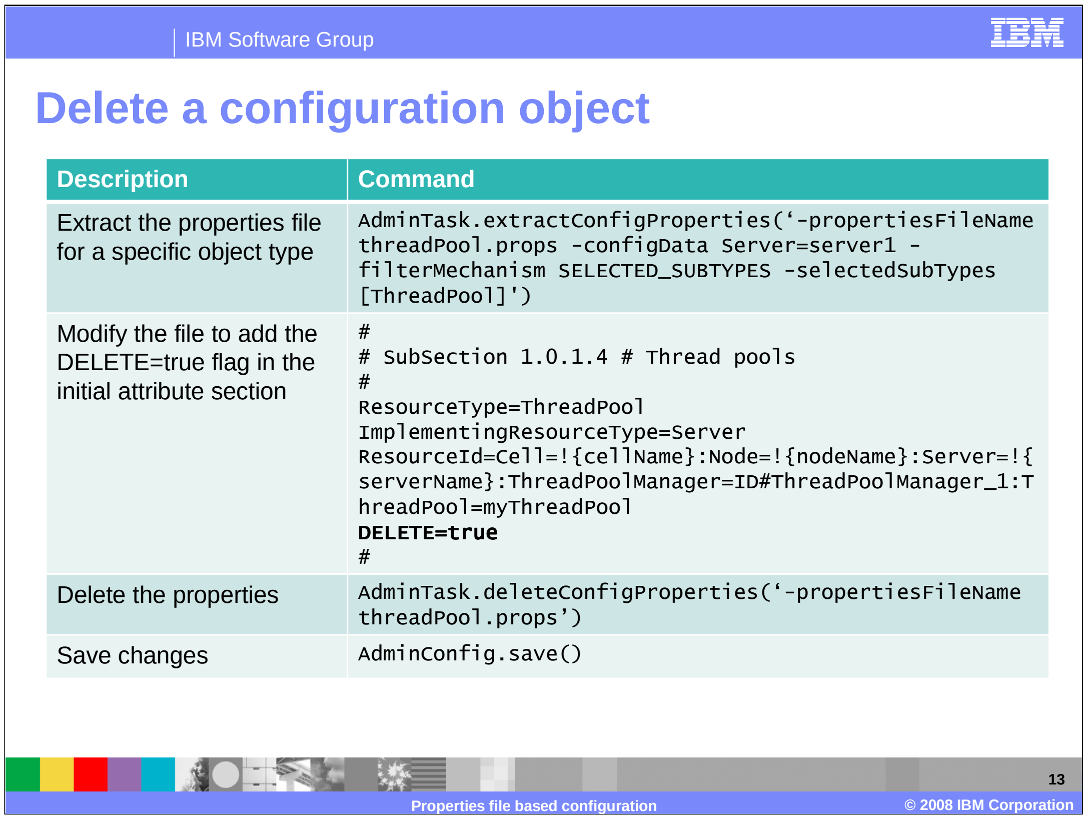
This section contains several examples of the properties file based configuration commands in action.



Modify existing server configuration

Description	Command
Extract the properties file for the configuration	`AdminTask.extractConfigProperties('-propertiesFileName server1.props -configData Server=server1')`
Modify the file	Use your favorite text editor to change properties in server1.props and save the changes to the file
Validate the updated properties file	`AdminTask.validateConfigProperties('-propertiesFileName server1.props')`
Apply the updated properties file to the configuration	`AdminTask.applyConfigProperties('-propertiesFileName server1.props')`
Save changes	`AdminConfig.save()`

One common scenario for working with your configuration using properties files is to extract a properties file based on your current environment, make modifications to the extracted file, and then apply the updated properties file to your configuration. You can use any text editor to modify the properties file. When you have made your changes, it is a good practice to validate the properties file before applying it to your configuration.



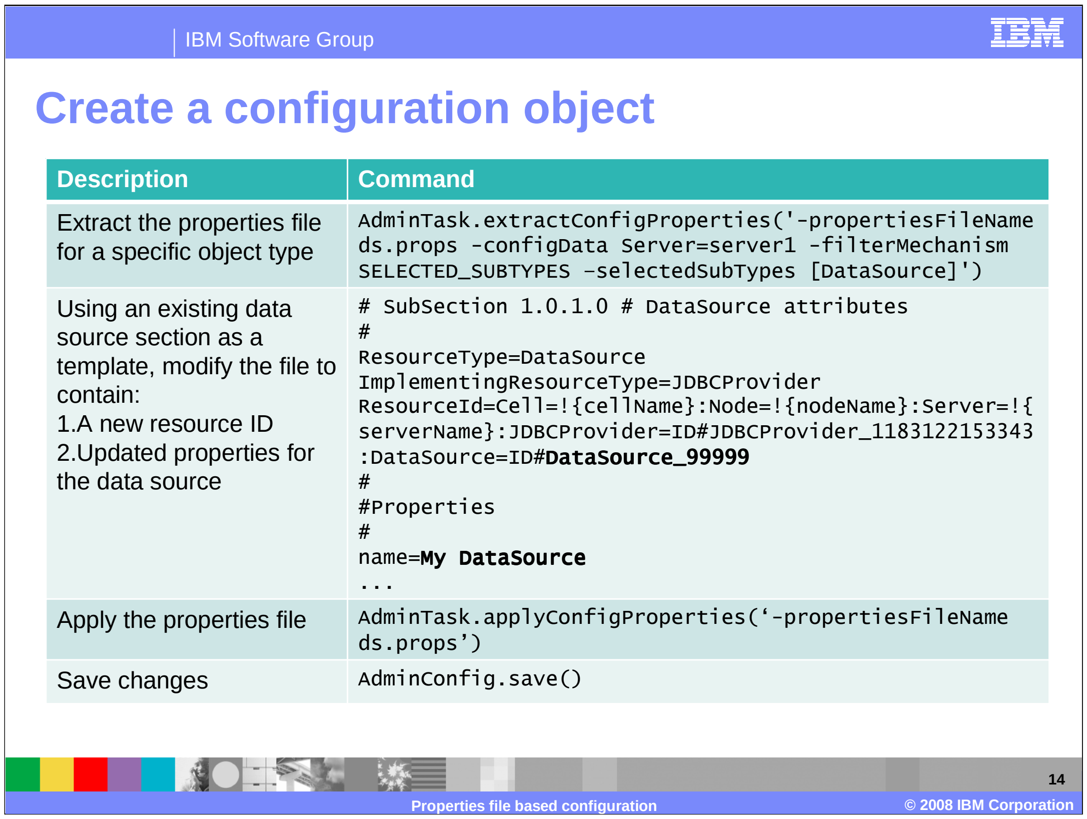
Delete a configuration object

Description	Command
Extract the properties file for a specific object type	`AdminTask.extractConfigProperties('-propertiesFileName threadPool.props -configData Server=server1 -filterMechanism SELECTED_SUBTYPES -selectedSubTypes [ThreadPool]')`

Modify the file to add the DELETE=true flag in the initial attribute section	`#`
	`# SubSection 1.0.1.4 # Thread pools`
	`#`
	`ResourceType=ThreadPool`
	`ImplementingResourceType=Server`
	`ResourceId=Cell=!{cellName}:Node=!{nodeName}:Server=!{serverName}:ThreadPoolManager=ID#ThreadPoolManager_1:ThreadPool=myThreadPool`
	`DELETE=true`
	`#`

Delete the properties	`AdminTask.deleteConfigProperties('-propertiesFileName threadPool.props')`
Save changes	`AdminConfig.save()`

Properties based configuration allows you to delete objects from your configuration. To delete a configuration object, you need a properties file that contains the resource information for that object. If you know what type of object you want to delete, you can extract a properties file for your server using an object type filter, as shown in the example on this page. When you have the properties file, you need to add the flag DELETE=true to the section that contains the resource identifier for the object you want to delete, and then run the deleteConfigProperties command on your properties file.



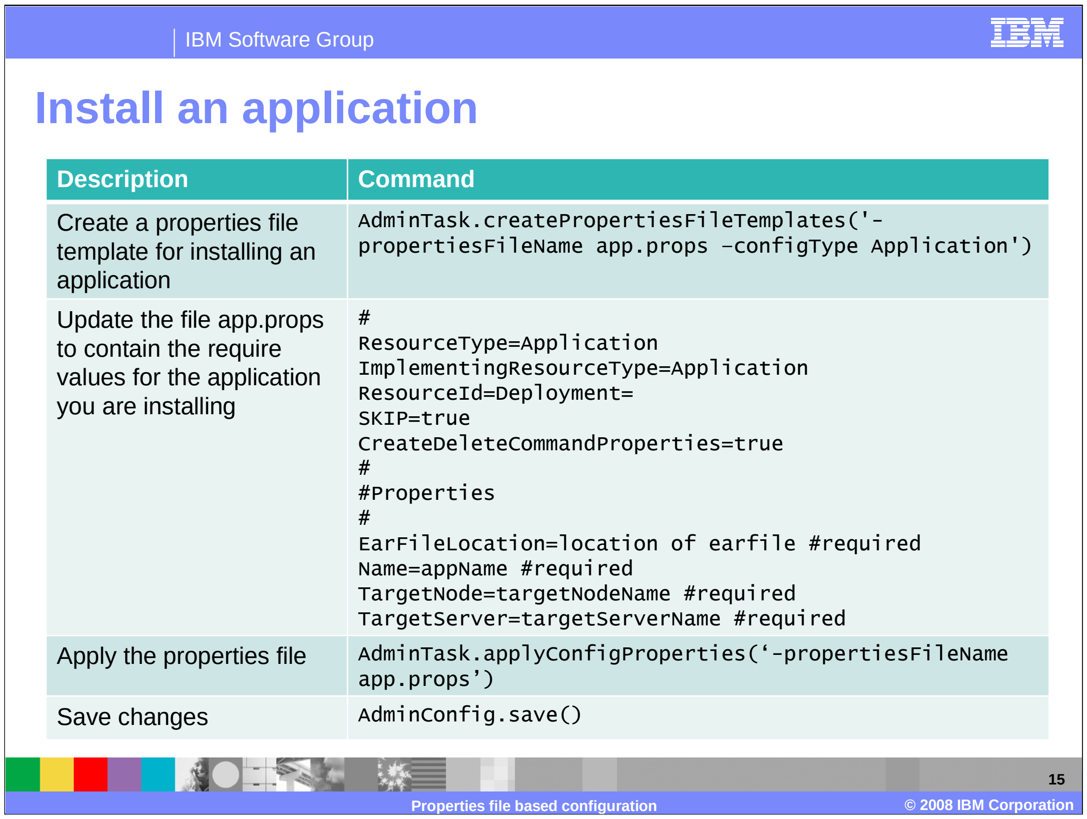
Create a configuration object

Description	Command
Extract the properties file for a specific object type	`AdminTask.extractConfigProperties('-propertiesFileName ds.props -configData Server=server1 -filterMechanism SELECTED_SUBTYPES -selectedSubTypes [DataSource]')`

Using an existing data source section as a template, modify the file to contain:
1.A new resource ID
2.Updated properties for the data source
	`# SubSection 1.0.1.0 # DataSource attributes`
	`#`
	`ResourceType=DataSource`
	`ImplementingResourceType=JDBCProvider`
	`ResourceId=Cell=!{cellName}:Node=!{nodeName}:Server=!{serverName}:JDBCProvider=ID#JDBCProvider_1183122153343:DataSource=ID#`**`DataSource_99999`**
	`#`
	`#Properties`
	`#`
	`name=`**`My DataSource`**
	`...`

Apply the properties file	`AdminTask.applyConfigProperties('-propertiesFileName ds.props')`
Save changes	`AdminConfig.save()`

When you apply a properties file that contains a new resource identifier that does not exist in the current configuration repository, then a new configuration object is created with that resource identifier. If you want to create a DataSource object, then extract a properties file for your server using the DataSource filter to create a template of the appropriate format. Find a section in the file that describes a data source and modify it to contain the properties that you want your data source to have. Be sure to update the resource identifier to some new value that does not exist in your configuration. Then, apply the properties file and save your changes. At that point, the new configuration object has been created in your configuration repository.



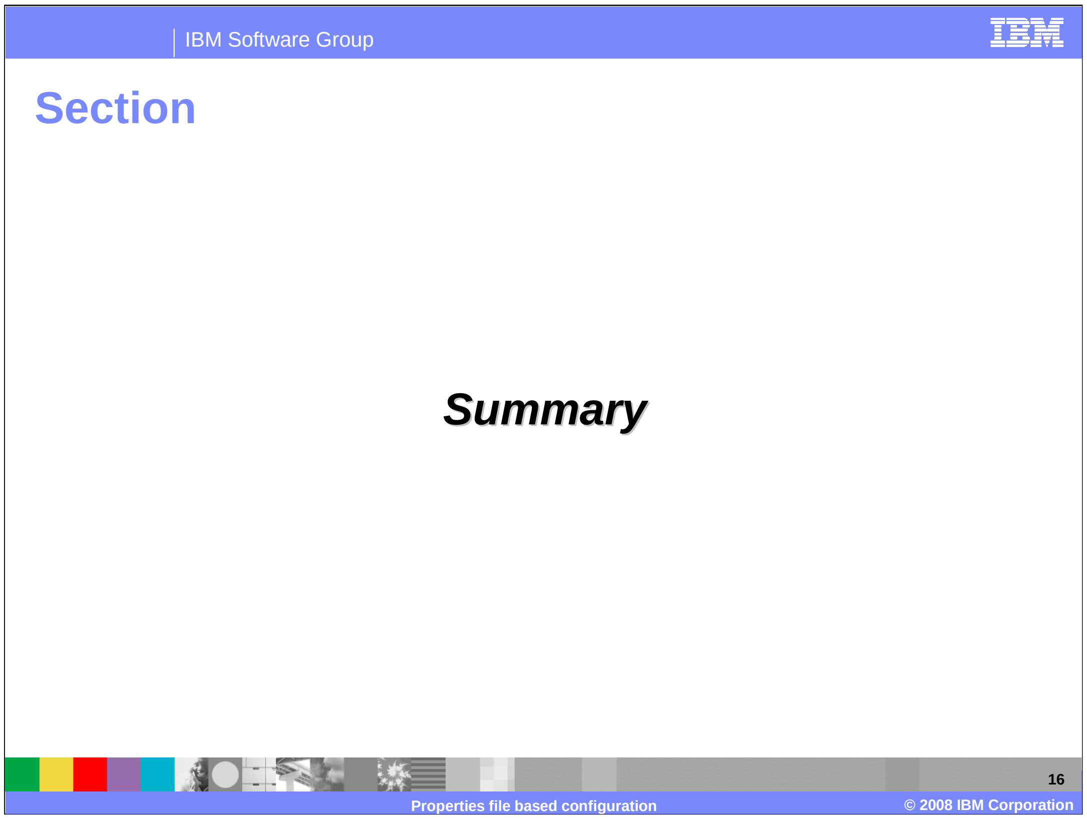
Install an application

Description	Command
Create a properties file template for installing an application	`AdminTask.createPropertiesFileTemplates('-propertiesFileName app.props –configType Application')`

Update the file app.props to contain the require values for the application you are installing	`#`
	`ResourceType=Application`
	`ImplementingResourceType=Application`
	`ResourceId=Deployment=`
	`SKIP=true`
	`CreateDeleteCommandProperties=true`
	`#`
	`#Properties`
	`#`
	`EarFileLocation=location of earfile #required`
	`Name=appName #required`
	`TargetNode=targetNodeName #required`
	`TargetServer=targetServerName #required`

Apply the properties file	`AdminTask.applyConfigProperties(‘-propertiesFileName app.props’)`
Save changes	`AdminConfig.save()`

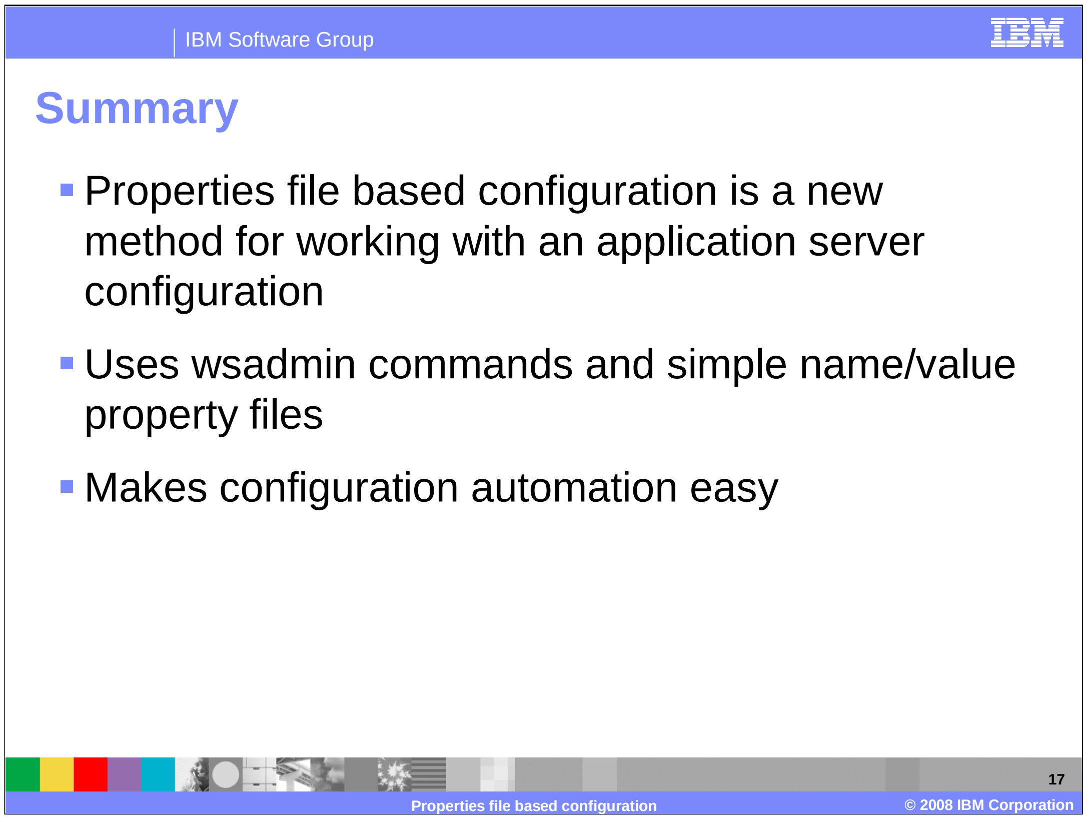
There are four different types of properties file templates that you can create with the createPropertiesFileTemplates command: Server, ServerCluster, Application, and AuthorizationGroup. The templates are properties files that contain the required parameters to create a configuration object of that type. So, when you create a template for the application configuration type, you need to provide a resource ID and information about the application that you are deploying. The properties file template contains comments and instructions for how to modify and use the template.



Section

Summary

This section contains a summary of the material covered in this presentation.


IBM Software Group

Summary

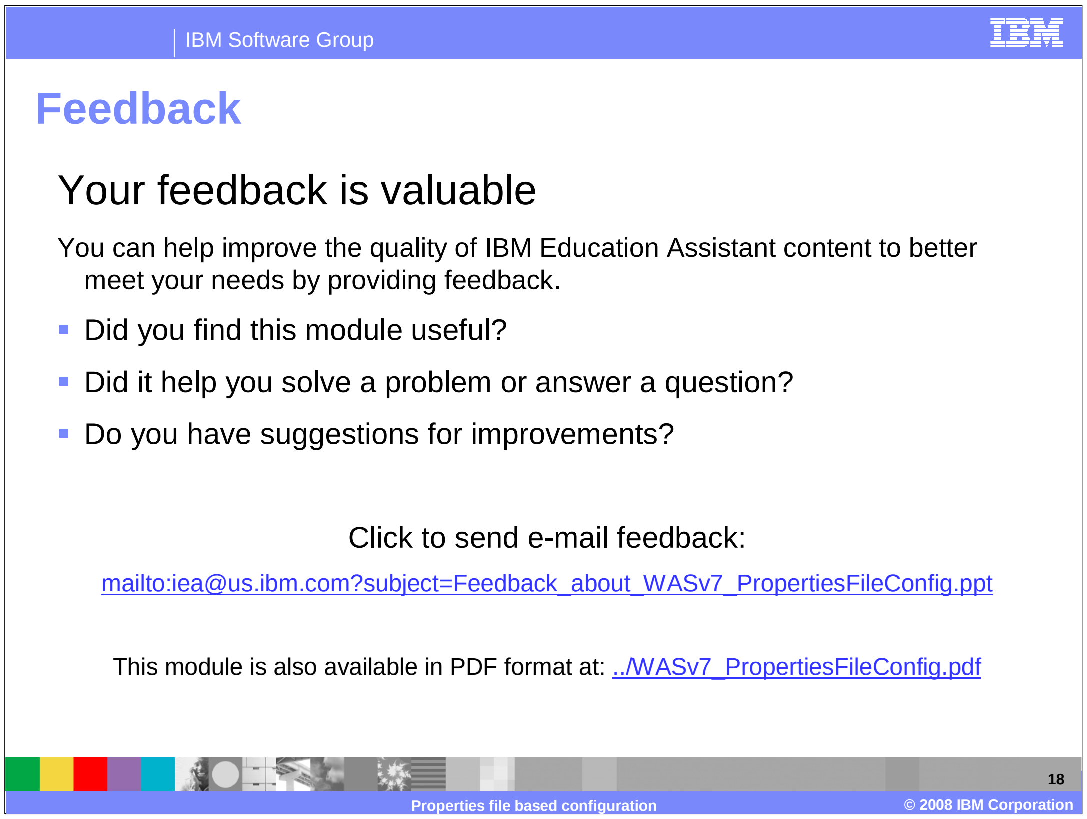
- Properties file based configuration is a new method for working with an application server configuration
- Uses wsadmin commands and simple name/value property files
- Makes configuration automation easy

17

Properties file based configuration

© 2008 IBM Corporation

Properties file based configuration is a new method for working with an application server configuration that has been introduced in WebSphere Application Server V7. This configuration scheme uses wsadmin commands and simple properties files to modify objects in the application server's configuration repository. Properties files make it easy to automate the process for configuring your environment without having to write complex scripts. They can be used in conjunction with configuration archives to replicate and customize application server configurations.


IBM Software Group

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_WASv7_PropertiesFileConfig.ppt

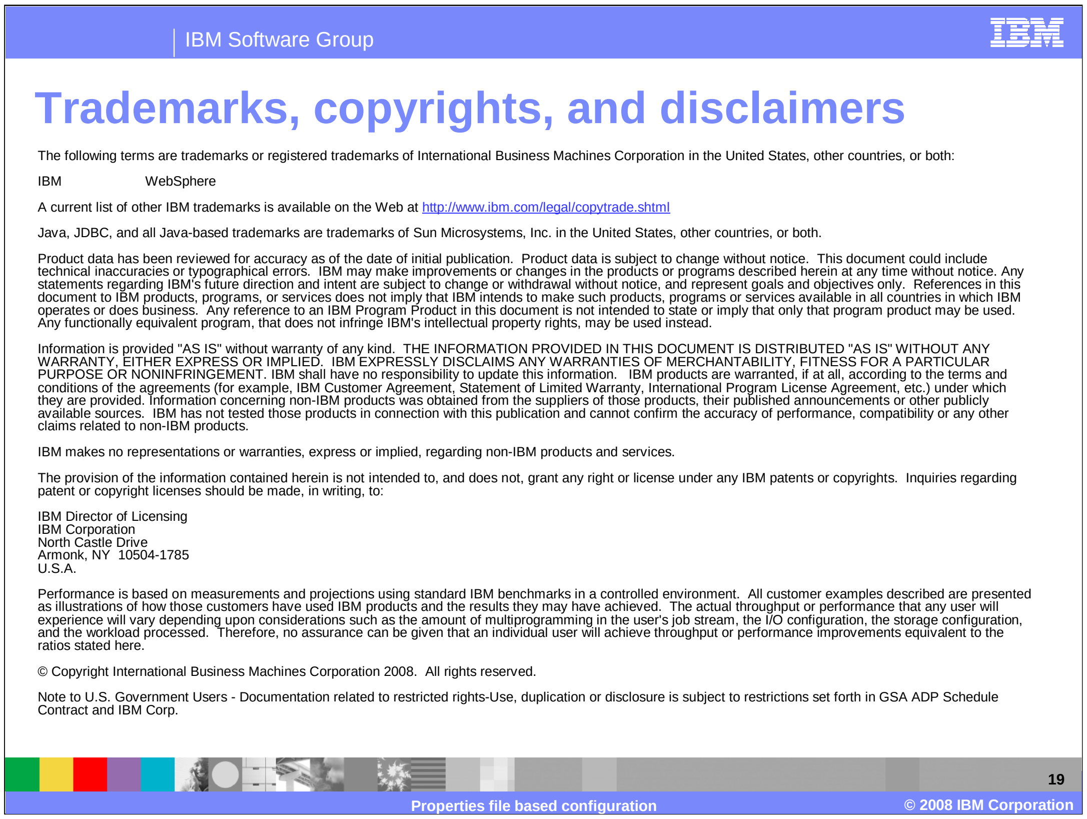
This module is also available in PDF format at: ../WASv7_PropertiesFileConfig.pdf

18

Properties file based configuration

© 2008 IBM Corporation

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



Trademarks, copyrights, and disclaimers