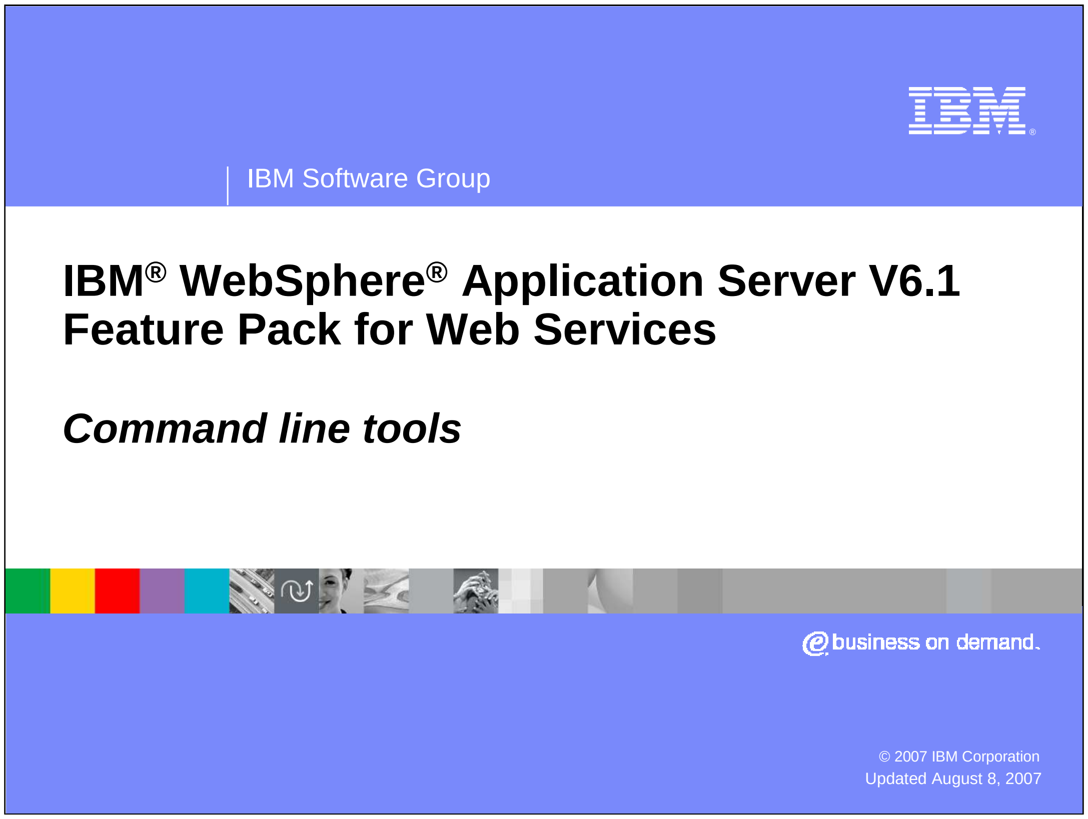

IBM Software Group

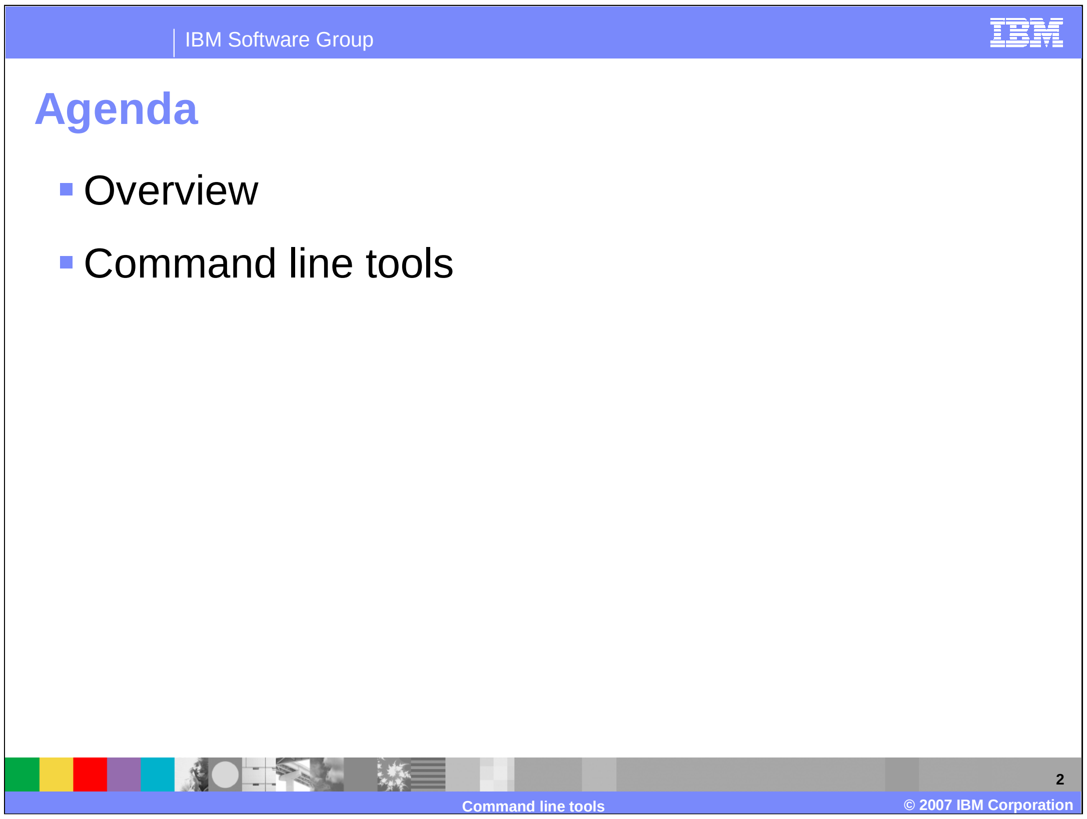
IBM® WebSphere® Application Server V6.1 Feature Pack for Web Services

Command line tools

business on demand.

Updated August 8, 2007

This presentation will focus on explaining the available command line tools that can be used for application development with the IBM WebSphere Application Server V6.1 Feature Pack for Web Services.



Agenda

- Overview
- Command line tools

This presentation will begin with an overview of the tools options available for developing Web Services applications for the Feature Pack for Web Services. It will then discuss the command line tools, or scripts that can be used to generate Web Services and related artifacts.

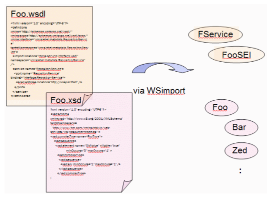
Section

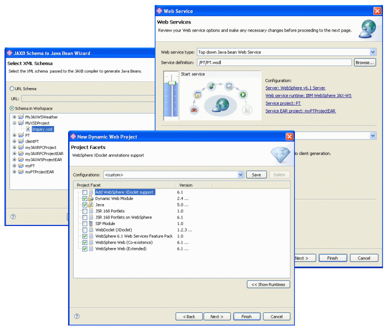
Overview

This section will provide an overview of the tools for the Feature Pack for Web Services.

Tools options

- Command-line tools
  - JAX-B 2.0 XSD->Java™ generation (xjc)
  - JAX-B 2.0-> Schema generation (schemagen)
  - JAX-WS 2.0 WSDL->Java (wsimport)
  - JAX-WS 2.0 Java->WSDL (wsgen)
- Rational® Application Developer and WebSphere Application Server toolkit
  - GUI Wizards to drive command-line tools
  - Feature pack awareness (facet)
  - Annotation validation
  - JAX-WS navigator view
  - Policy set support





For the application developer, the Feature Pack for Web Services provides a set of tools to help make development easier. At the most basic level there is command line tools to generate various artifacts. An XJC command can be used to generate Java artifacts based on a JAXB 2.0 XSD definition or also from a WSDL file. There are also WSIMPORT and WSGEN commands for top down and bottom up Web Services development. There are also updates to the AST and Rational Application Developer, with additions to wizards for the creation of Web Services. These have been extended to support JAX-WS and JAXB based Web Services with; annotation validation, graphical wizards, publishing tools for developed Web Services and a Jython debugger for scripting.

IBM Software Group

Section

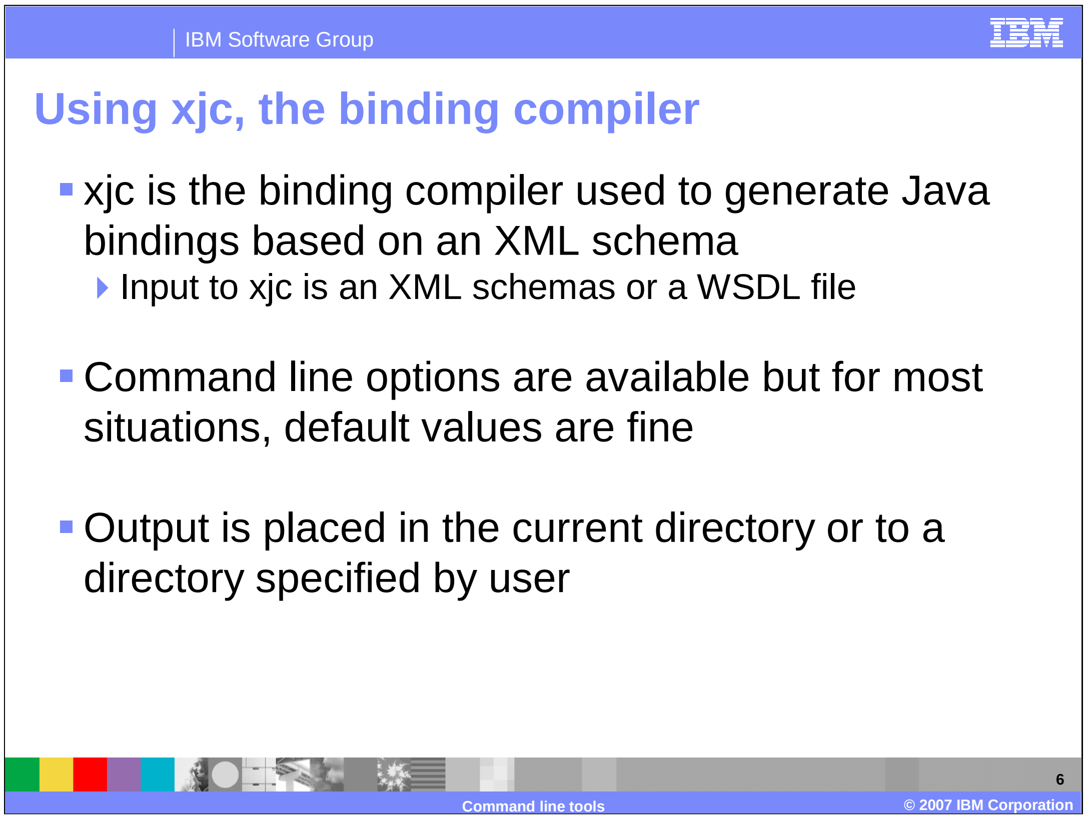
Command line tools

5

Command line tools

© 2007 IBM Corporation

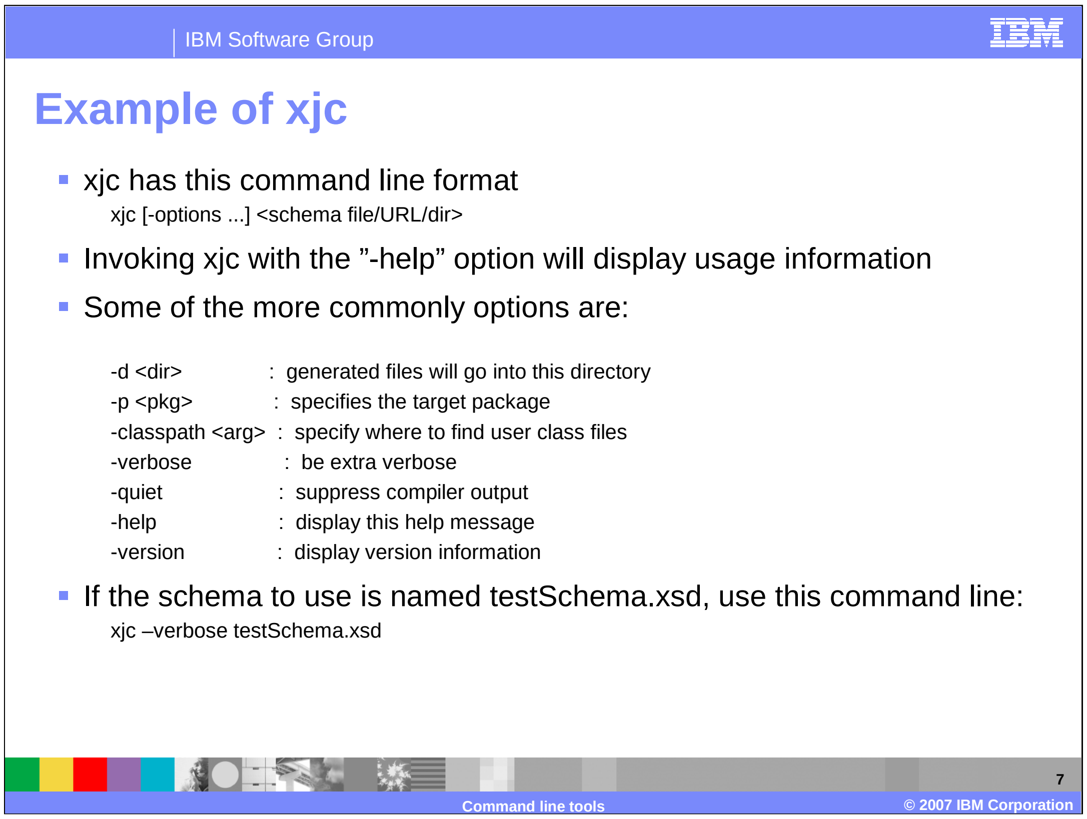
The next section discusses the command line tools for the Feature Pack for Web Services.



Using xjc, the binding compiler

- xjc is the binding compiler used to generate Java bindings based on an XML schema
  - Input to xjc is an XML schemas or a WSDL file
- Command line options are available but for most situations, default values are fine
- Output is placed in the current directory or to a directory specified by user

The XJC command can be used to compile the Java bindings based on an XML schema. XML schemas describe the data elements and relationships in an XML document. After a data mapping or binding exists, XML documents can be converted to and from Java objects. Generate fully annotated Java classes from an XML schema file by using the XJC command-line tool. The schema compiler is located in the bin directory under the application server root. The output will be placed in the current directory or one specified by you.



Example of xjc

- xjc has this command line format

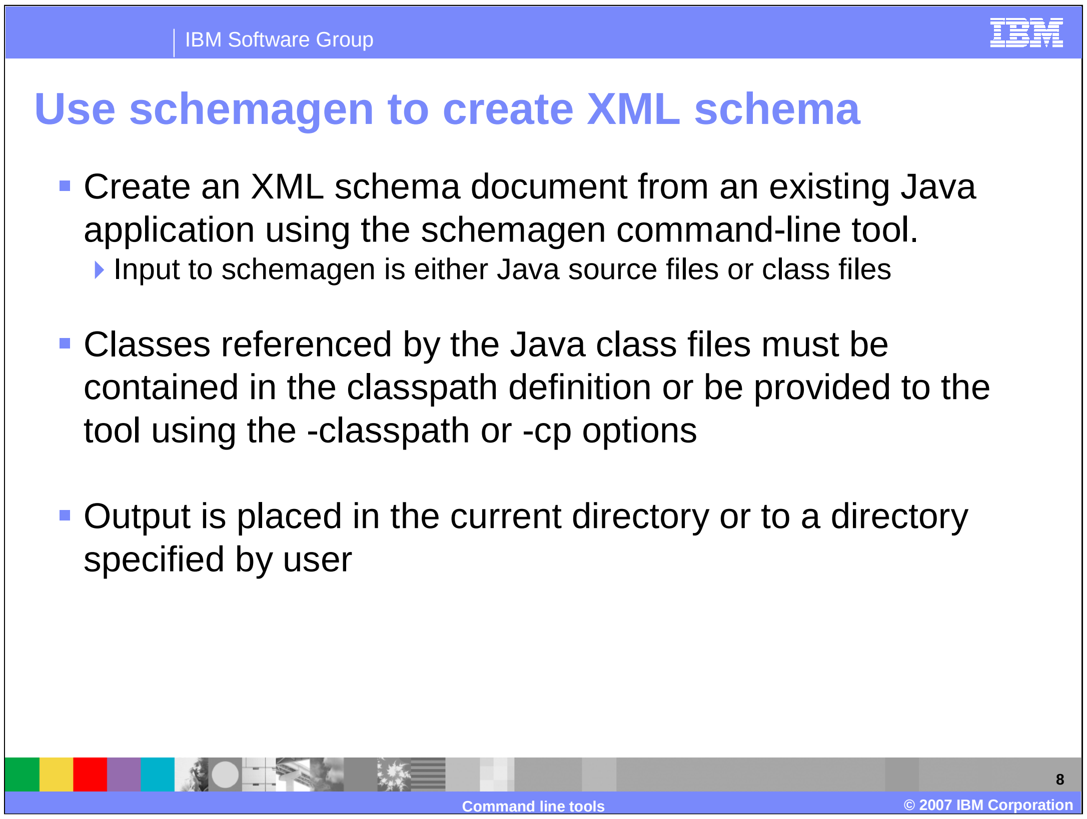
  xjc [-options ...] <schema file/URL/dir>
- Invoking xjc with the ”-help” option will display usage information
- Some of the more commonly options are:

-d <dir>	: generated files will go into this directory
-p <pkg>	: specifies the target package
-classpath <arg>	: specify where to find user class files
-verbose	: be extra verbose
-quiet	: suppress compiler output
-help	: display this help message
-version	: display version information

- If the schema to use is named testSchema.xsd, use this command line:

  xjc –verbose testSchema.xsd

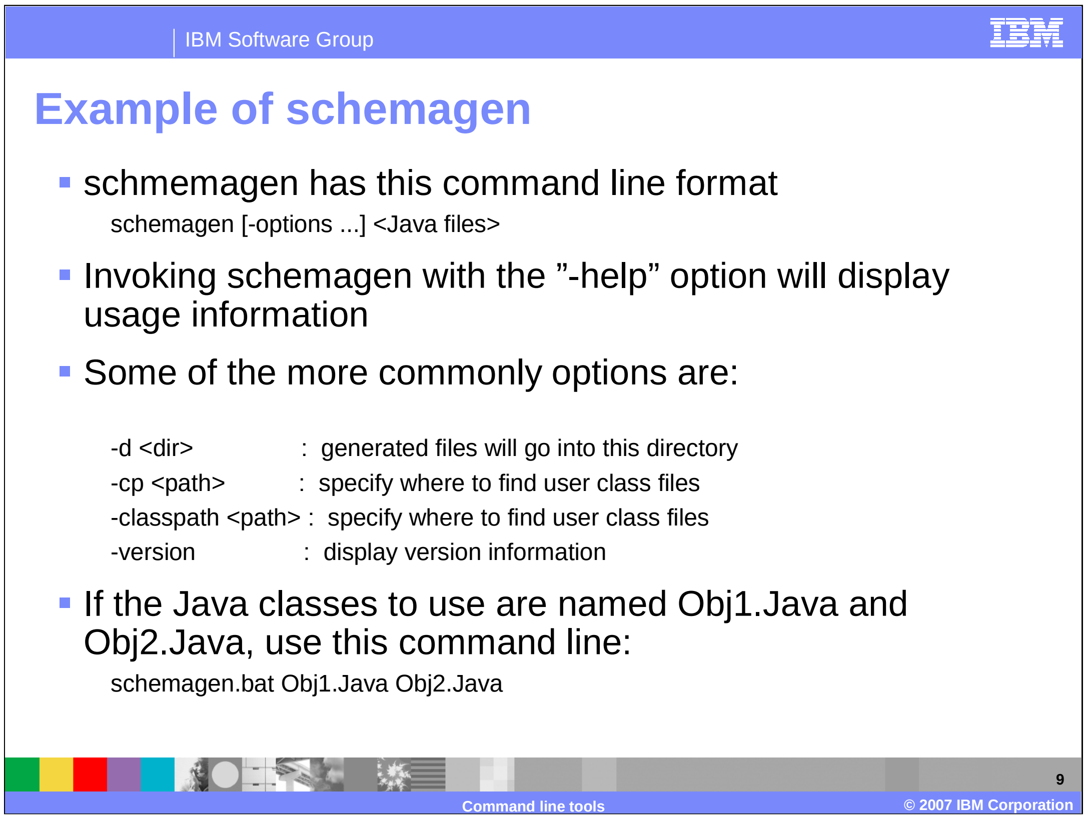
The above information shows how to use the XJC command. Use the –help option to display usage information on the command line. An example is given with a sample test schema.



Use schemagen to create XML schema

- Create an XML schema document from an existing Java application using the schemagen command-line tool.
  - Input to schemagen is either Java source files or class files
- Classes referenced by the Java class files must be contained in the classpath definition or be provided to the tool using the -classpath or -cp options
- Output is placed in the current directory or to a directory specified by user

An XML schema can be documented from existing Java classes, which represent the data elements of an application, using the JAXB schema generator or schemagen command-line tool. The JAXB schema generator processes either Java source files or class files. Annotations within the Java classes provide the capability to customize the default mappings from existing Java classes to the generated schema components. The XML schema file and the annotated Java class files contain all the necessary information that JAXB requires to parse the XML documents for serialization and deserialization.



Example of schemagen

- schmemagen has this command line format

  schemagen [-options ...] <Java files>

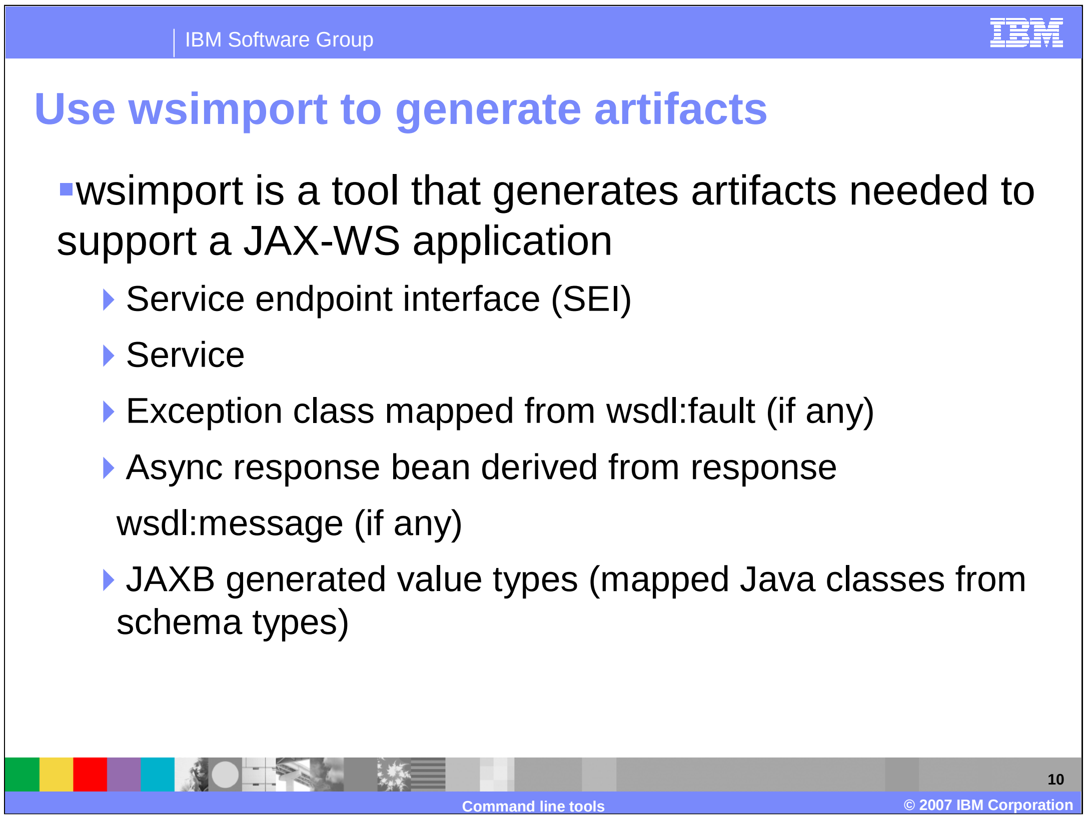
- Invoking schemagen with the ”-help” option will display usage information
- Some of the more commonly options are:

```
-d <dir>           :  generated files will go into this directory
-cp <path>         :  specify where to find user class files
-classpath <path> :  specify where to find user class files
-version           :  display version information
```

- If the Java classes to use are named Obj1.Java and Obj2.Java, use this command line:

  schemagen.bat Obj1.Java Obj2.Java

The above information shows how to use the schemagen command. Use the –help option to display usage information on the command line. An example is given with a sample test files.



IBM Software Group

Use wsimport to generate artifacts

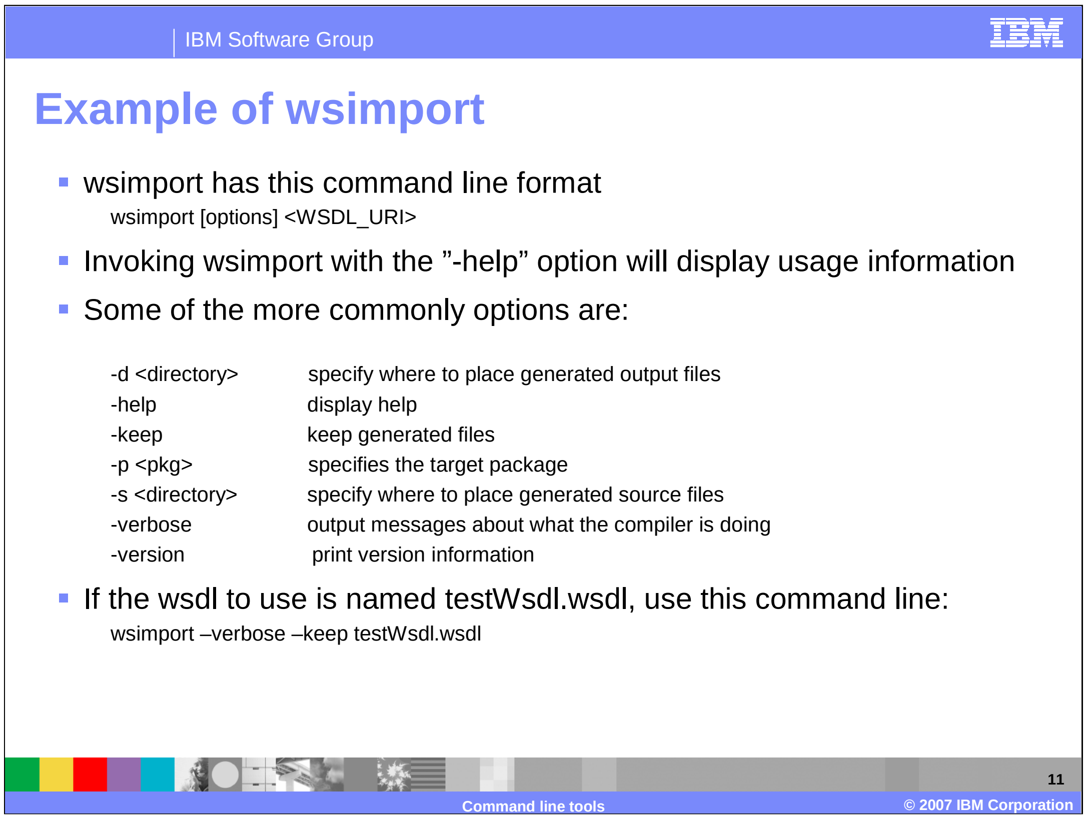
- wsimport is a tool that generates artifacts needed to support a JAX-WS application
  - Service endpoint interface (SEI)
  - Service
  - Exception class mapped from wsdl:fault (if any)
  - Async response bean derived from response wsdl:message (if any)
  - JAXB generated value types (mapped Java classes from schema types)

10

Command line tools

© 2007 IBM Corporation

The wsimport command-line tool processes an existing Web Services Description Language file and creates the required portable artifacts for developing JAX-WS based Web Service applications. The wsimport command-line tool supports the top-down approach to developing JAX-WS Web Services when a WSDL is used to generate the various artifacts, including the service endpoint interface, the service class, an exception class defined by the WSDL fault element, an asynchronous response bean based on the WSDL message element, and the JAXB generated types.



Example of wsimport

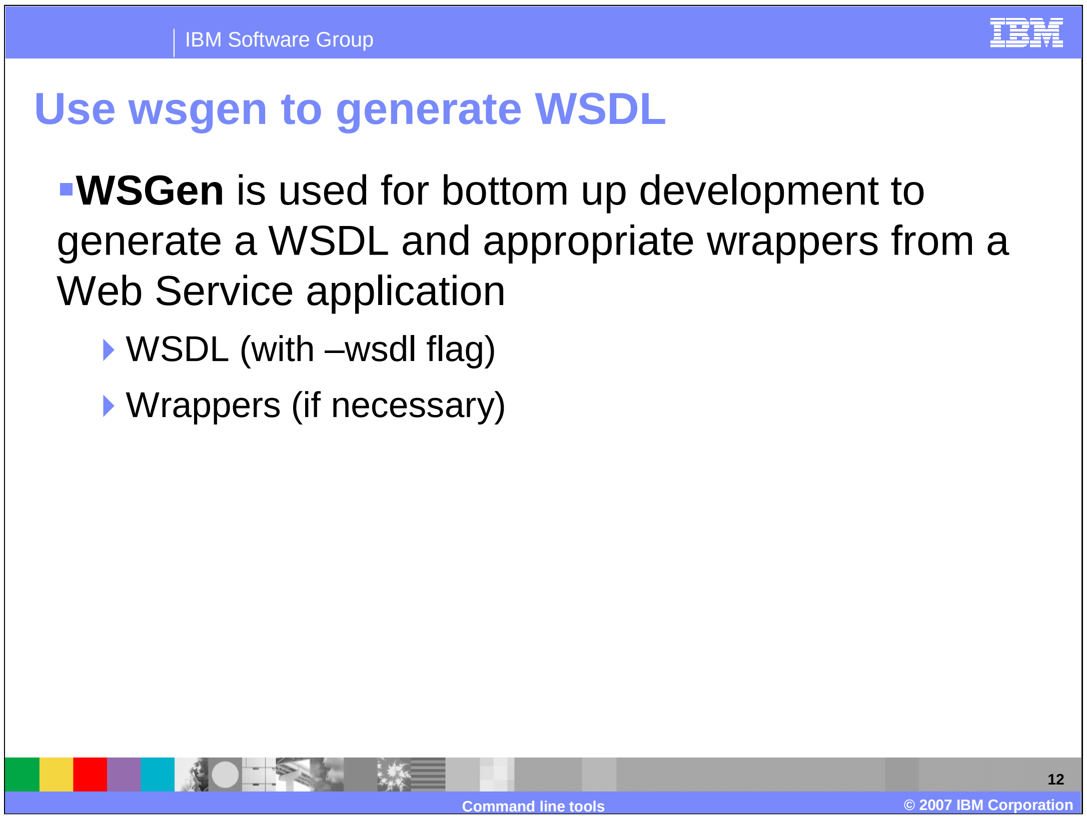
- wsimport has this command line format
  wsimport [options] <WSDL_URI>
- Invoking wsimport with the ”-help” option will display usage information
- Some of the more commonly options are:

-d <directory>	specify where to place generated output files
-help	display help
-keep	keep generated files
-p <pkg>	specifies the target package
-s <directory>	specify where to place generated source files
-verbose	output messages about what the compiler is doing
-version	print version information

- If the wsdl to use is named testWsdl.wsdl, use this command line:

```
wsimport –verbose –keep testWsdl.wsdl
```

The above information shows how to use the wsimport command. Use the –help option to display usage information on the command line. An example is given with a sample WSDL, notice the use of the –keep option to retain the generated files.


IBM Software Group

Use wsgen to generate WSDL

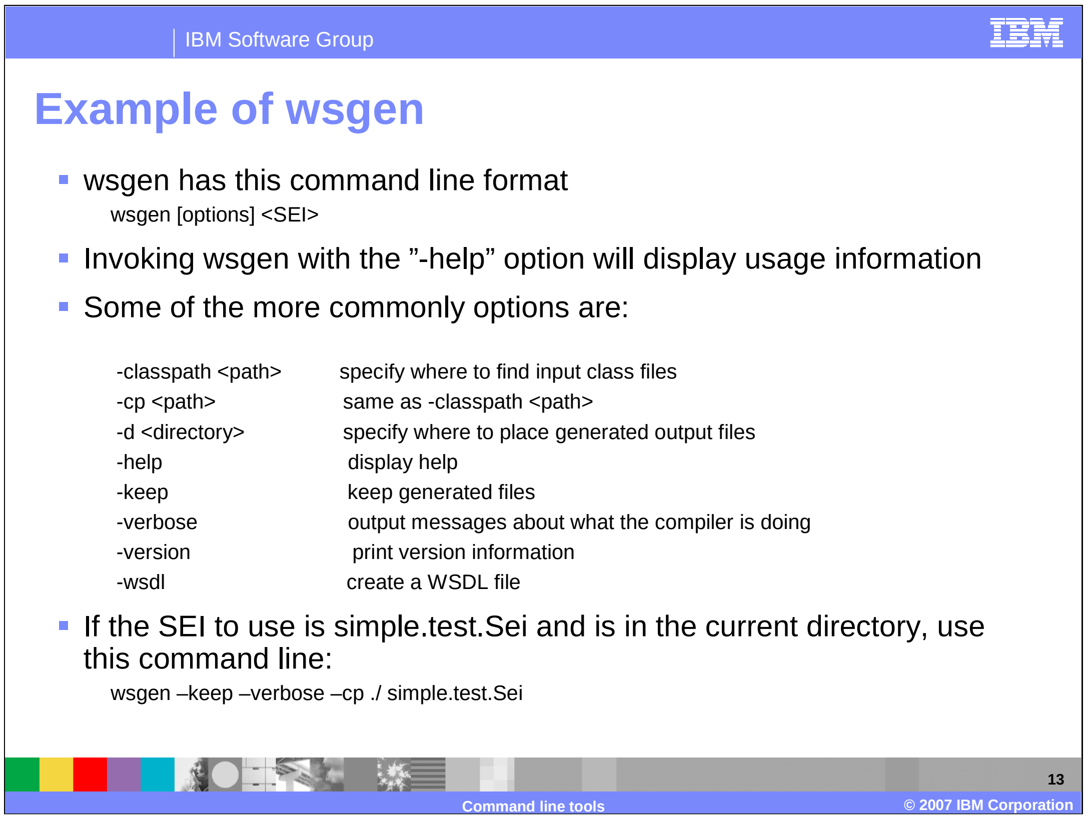
- **WSGen** is used for bottom up development to generate a WSDL and appropriate wrappers from a Web Service application
  - WSDL (with –wsdl flag)
  - Wrappers (if necessary)

12

Command line tools

© 2007 IBM Corporation

The wsgen command-line tool generates the necessary portable artifacts required for JAX-WS applications when starting from Java code. This tool will generate a WSDL file only when specified. When using a bottom-up approach to develop JAX-WS Web Services, creating a Web Service from a service endpoint implementation, use the **wsgen** tool to generate the required artifacts. The wsgen tool accepts a properly annotated service endpoint implementation using the @WebService annotation as input and generates artifacts for the WSDL and the JAXB wrappers that may be necessary.



Example of wsgen

- wsgen has this command line format

  wsgen [options] <SEI>
- Invoking wsgen with the ”-help” option will display usage information
- Some of the more commonly options are:

-classpath <path>	specify where to find input class files
-cp <path>	same as -classpath <path>
-d <directory>	specify where to place generated output files
-help	display help
-keep	keep generated files
-verbose	output messages about what the compiler is doing
-version	print version information
-wsdl	create a WSDL file

- If the SEI to use is simple.test.Sei and is in the current directory, use this command line:

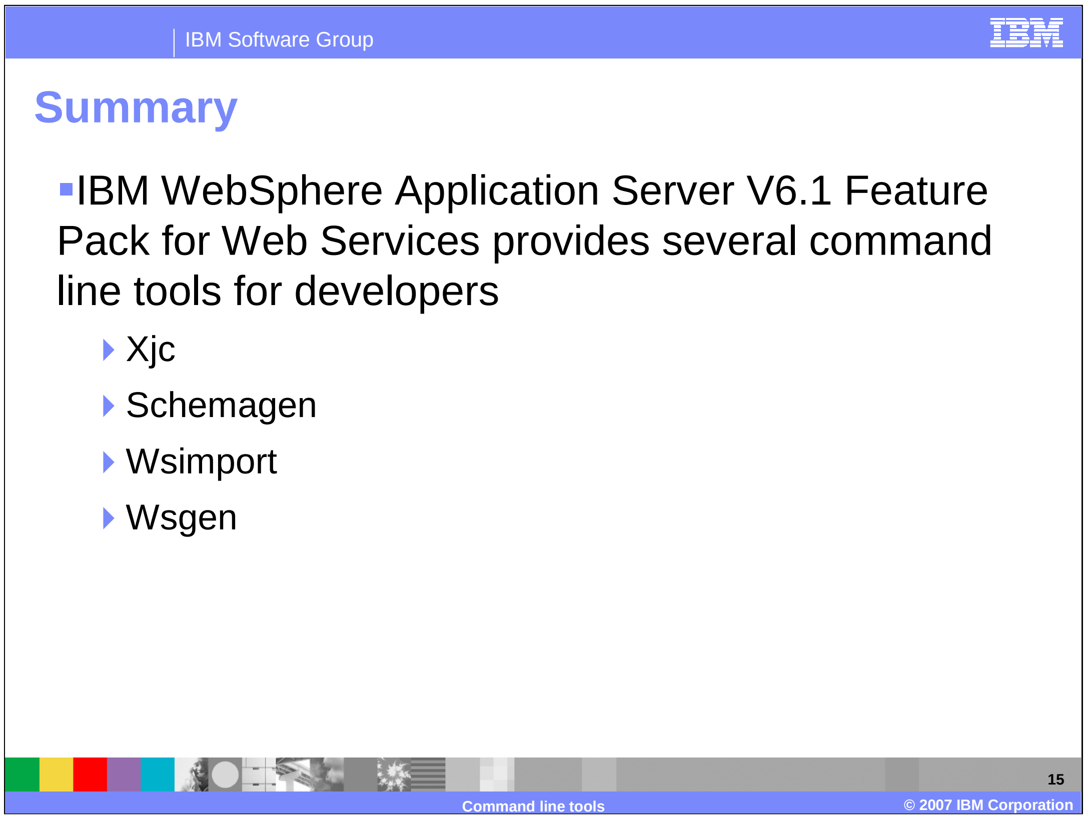
```
wsgen –keep –verbose –cp ./ simple.test.Sei
```

The above information shows how to use the wsgen command. Use the –help option to display usage information on the command line. An example is given with a sample service endpoint interface, notice the use of the –keep option to retain the generated files.

Section

Summary

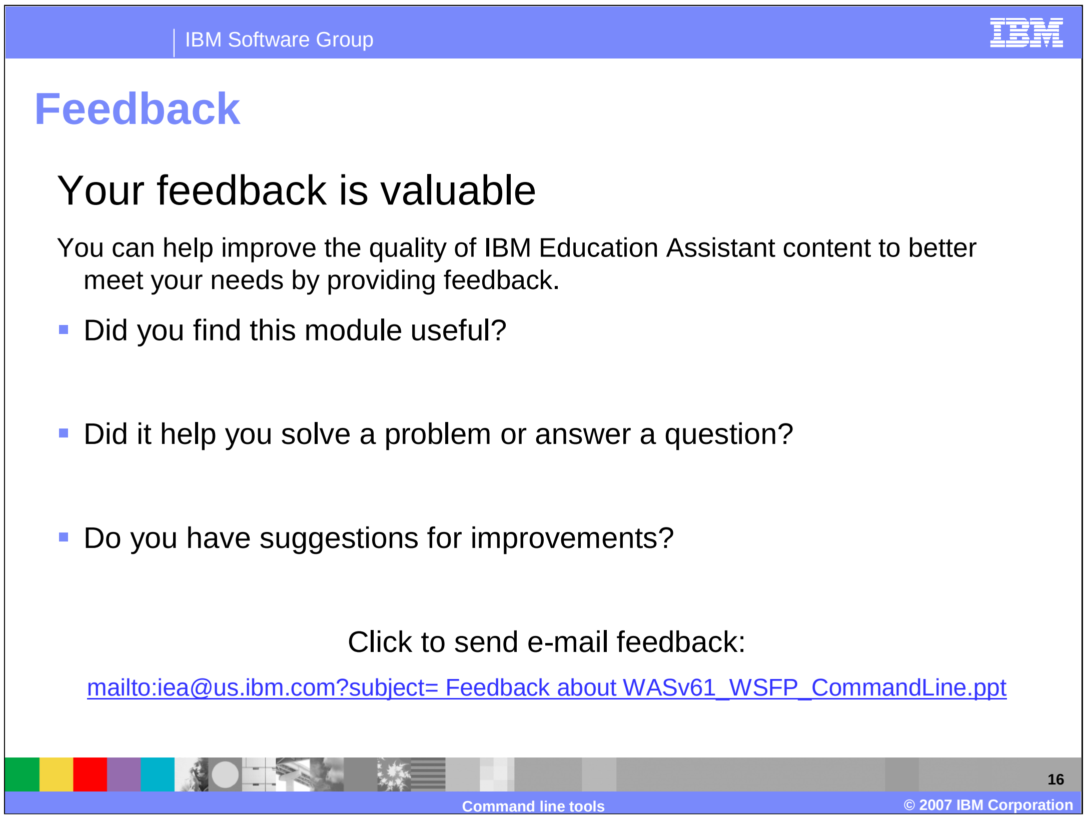
The next section will summarize the materials.



Summary

- IBM WebSphere Application Server V6.1 Feature Pack for Web Services provides several command line tools for developers
  - Xjc
  - Schemagen
  - Wsimport
  - Wsgen

Developers have a number of choices when creating JAX-WS applications with the Feature Pack for Web Services. Command line tools are available to supplement the Application Server toolkit and IBM Rational Application Developer. This presentation explained the command line tools in detail.



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:

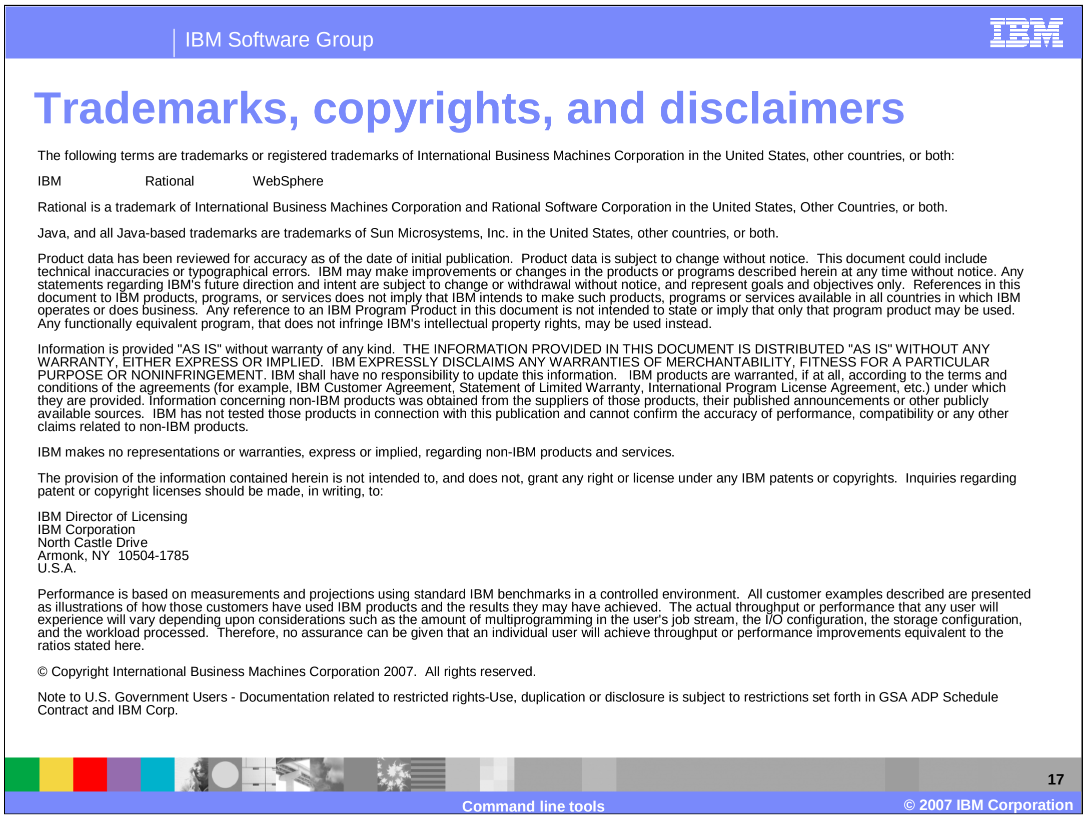
mailto:iea@us.ibm.com?subject= Feedback about WASv61_WSFP_CommandLine.ppt

16

Command line tools

© 2007 IBM Corporation

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



Trademarks, copyrights, and disclaimers