Command Line Tools

Table of Contents

The Celtix wsdl2java Utility

Using the `wsdl2java` utility gives you greater control over the code generation process. The syntax of the command is summarized in CodeSnap 6.

```
wsdl2java -p <[wsdl-namespace =]Package-Name>* -b <binding-name>*
           -d <output-directory> -compile -classdir <compile-classes-directory>
           -client -server -impl -all -ant
           -nexclude <schema-namespace [= java-packagename]>*
           -exsh <(true, false)> -v -verbose -quiet <WSDLfile>
```

CodeSnap 6 `wsdl2java` *syntax.*

The arguments used to manage code generation are reviewed in the following table.

Command Line Argument	***Interpretation***

`-?`
`-help`
`-h`
	Displays the help for this utility

`-p <[wsdl-namespace=]Package-name>*`	Specifies zero, or more, package names to use for the generated code. Optionally specifies the WSDL namespace to package name mapping.
`-b <binding-file-name>*`	Specifies zero, or more, JAXWS or JAXB binding files. Use spaces to separate multiple entries.
`-d <output-directory>`	Specifies the directory into which the generated code is placed.
`-compile`	Compiles generated Java files.
`-classdir`	Specifies the directory into which the compiled classes are placed.
`-client`	Generates starting point code for the client mainline.
`-server`	Generates starting point code for the server mainline.

Command Line Argument	Interpretation
`-impl`	Generates starting point code for the implementation object.
`-all`	Generates all starting point code: types, service proxy, service interface, server mainline, client mainline, implementation object, Ant `build.xml` file.
`-ant`	Generates the Ant `build.xml` file.
`-nexclude <schema-namespace [= java-packagename]>`	Ignores the specified WSDL schema namespace when generating code. This option can be specified multiple times. Also, optionally specifies the Java package name to use for this wsdl type namespace.
`-exsh <(true/false)>`	Enables or disables processing of extended soap header message binding.
`-v`	Displays the version number for the tool.
`-verbose`	Displays comments during the code generation process.
`-quiet`	Suppresses comments during the code generation process.

You must specify the absolute or relative path to the WSDL file as the last argument. All other arguments are optional and may be listed in any order.

The Celtix java2wsdl Utility

The **`java2wsdl`** tool generates the WSDL used in Celtix development and deployment. This tool ues a Web service endpoint's implementation class and types classes to generate WSDL. The syntax of the command is summarized in CodeSnap 7.

```
java2wsdl -o <output-file> -cp <class-path> -t <target-namespace>
          -servicename <service-name> -h -v -verbose -quiet <classname>
```

CodeSnap 7 `java2wsdl` *syntax.*

The arguments used to manage code generation are reviewed in the following table.

Command Line Argument	Interpretation

`-?`
`-help`
`-h`
	Displays the online help for this utility

`-o <output-file>`	Specifies the generated WSDL file name. By default the tool uses the “” to name the generated WSDL file.
`-cp <class-path>`	Specify the SEI and types class search path of directories and zip/jar files

Command Line Argument	Interpretation
-t *<target-namespace>*	Specifies the target namespace to use for the generated WSDL.
-servicename *<service-name>*	Specifies the value of the generated wsdl:service element's name attribute.
<classname>	Specifies the name of the service endpoint class
-v	Displays the version number for the tool.
-verbose	Displays comments during the code generation process.
-quiet	Suppresses comments during the code generation process.

You must include the *classname* argument. All other arguments are optional and may be listed in any order. This tool will search and load the service endpoint class and types class .Make sure these classes are on your CLASSPATH or provided by -cp flag.

The Celtix wsdl2soap Utility

This tool will generate a new WSDL file with SOAP binding from an existing WSDL portType element from.The syntax of the command is summarized in CodeSnap 8.

```
wsdl2soap -i <port-type-name> -b <binding-name>
           -d <output-directory> -o <output-file>
           -n <soap-body-namespace> -style <(document/rpc)>
           -use <(literal/encoded)> -h -v -verbose -quiet <wsdlurl>
```

CodeSnap 8 wsdl2soap *syntax.*

The arguments used to manage code generation are reviewed in the following table.

Command Line Argument	Interpretation
-? -help -h	Displays the online help for this utility
-i *<port-type-name>*	Specifies the portType element to use.
-b *<binding-name>*	Specifies the name of the generated SOAP binding.
-d *<output-directory>*	Specifies the directory to place generated WSDL file.
-o *<output-file>*	Specifies the name of the generated WSDL file.
-n *<soap-body-namespace>*	Specifies the SOAP body namespace when the style is RPC.
-style <(document/rpc)>	Specifies the encoding style (Document/RPC) to use in the SOAP binding. The default is document.

Command Line Argument	Interpretation
`-use <(literal/encoded)>`	Specifies the binding use (encoded/literal) to use in the SOAP binding. The default is literal.
`-v`	Displays the version number for the tool.
`-verbose`	Displays comments during the code generation process.
`-quiet`	Suppresses comments during the code generation process.

The `-i` *`<port-type-name>`* and the *`<wsdlurl>`* arguments are required. If the `-style rpc` argument is specified, the `-n` *`<soap-body-namspace>`* argument is also required. All other arguments are optional and may be listed in any order.

The Celtix wsdl2service Utility

The **`wsdl2service`** tool generates an HTTP or a JMS service from an existing WSDL document. The syntax of the command is summarized in CodeSnap 9.

```
wsdl2service -transport <http/jms> -e <service-name>
             -p <port-name> -n <attribute-binding-name>
             [-a <address>] -o <output-file>
             -d <output-directory> -h -v -verbose -quiet
             <wsdlurl>
```

CodeSnap 9 `wsdl2service`*syntax.*

The arguments used to manage code generation are reviewed in the following table.

Command Line Argument	Interpretation

`-?`
`-help`
`-h`
	Displays the online help for this utility.

`-transport <http/jms>`	Specifies the type of transport to use for the generated service.
`-e` *`<service-name>`*	Specifies the value of the generated `service` element's `name` attribute.
`-p` *`<port-name>`*	Specifies the value of the generated `port` element's `name` attribute. To specify multiple `port` elements, separate the names by a space.
`-a` *`<address>`*	Specifies the value used in the `address` element of the port.
`-n` *`<attribute-binding-name>`*	Specifies the binding used to generate the service.

Command Line Argument	Interpretation
`-o <output-file>`	Specifies the name of the generated WSDL file.
`-d <output-directory>`	Specifies the directory in which the generated WSDL is placed.
`-v`	Displays the version number for the tool.
`-verbose`	Displays comments during the code generation process.
`-quiet`	Suppresses comments during the code generation process.

You must specify the `-n <attribute-binding-name>` argument and the `wsdlurl` argument. All other arguments are optional and may be listed in any order.

The Celtix wsdl2xml Utility

The **`wsdl2xml`** tool generates an XML binding from an existing WSDL document. The syntax of the command is summarized in CodeSnap 10.

```
wsdl2xml -i <port-type-name> -b <binding-name> -e <service-name>
         -p <port-name> -a <address> -d <output-directory>
         -o <output-file> -h -v -verbose -quiet
         <wsdlurl>
```

CodeSnap 10 `wsdl2xml`*syntax.*

The arguments used to manage code generation are reviewed in the following table.

Command Line Argument	Interpretation

`-?`
`-help`
`-h`
	Displays the online help for this utility

`-i <port-type-name>`	Specifies the `portType` element to use.
`-b <binding-name>`	Specifies the name of the generated SOAP binding.
`-e <service-name>`	Specifies the value of the generated `service` element's `name` attribute.
`-p <port-name>`	Specifies the value of the generated `port` element's `name` attribute. To specify multiple port elements, separate the names by a space.
`-a <address>`	Specifies the value used in the `address` element of the generated `port` element.

Command Line Argument	Interpretation
`-d <output-directory>`	Specifies the directory to place generated WSDL file.
`-o <output-file>`	Specifies the name of the generated WSDL file.
`-v`	Displays the version number for the tool.
`-verbose`	Displays comments during the code generation process.
`-quiet`	Suppresses comments during the code generation process.

The `-i <port-type-name>` and the `<wsdlurl>` arguments are required. All other arguments are optional except and may be listed in any order.

The Celtix xsd2wsdl Utility

The **`xsd2wsdl`** tool imports an XMLSchema document and generates a WSDL document containing a `types` element populated by the types defined in the XMLSchema document. The syntax of the command is summarized in CodeSnap 11.

```
xsd2wsdl -t <target-name-space> -n <wsdl-name>
         -d <output-directory> -o <output-file>
         -h -v -verbose -quiet <xsdurl>
```

CodeSnap 11 `xsd2wsdl` *syntax.*

The arguments used to manage code generation are reviewed in the following table.

Command Line Argument	Interpretation

`-?`
`-help`
`-h`
	Displays the online help for this utility.

`-t <target-name-space>`	Specifies the target namespace for the generated WSDL.
`-n <wsdl-name>`	Specifies the value of the generated `wsdl:definition` element's `name` attribute.
`-d <output-directory>`	Specifies the directory in which the generated WSDL is placed.
`-o <output-file>`	Specifies the name of the generated WSDL file.
`-v`	Displays the version number for the tool.
`-verbose`	Displays comments during the code generation process.

Command Line Argument	*Interpretation*
`-quiet`	Suppresses comments during the code generation process.

The `-t` *`<target-name-space>`* and the *`<wsdlurl>`* arguments are required. All other arguments are optional and may be listed in any order.