

Command Line Tools

Table of Contents

The Celtix wsdl2java Utility.....	1
The Celtix java2wsdl Utility.....	2
The Celtix wsdl2soap Utility.....	3
The Celtix wsdl2service Utility.....	4
The Celtix wsdl2xml Utility.....	5
The Celtix xsd2wsdl Utility.....	6
The Celtix wsdlvalidator Utility.....	6

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 the following illustration.

```
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)> -validate -v -verbose -quiet <wsdlfile>
```

The arguments used to manage the code generation process 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 files are written.
-compile	Compiles generated Java files.
-classdir	Specifies the directory into which the compiled class files are written.
-client	Generates starting point code for a client mainline.
-server	Generates starting point code for a server mainline.
-impl	Generates starting point code for an implementation object.

The Celtix wsdl2java Utility: The Celtix wsdl2java Utility

Command Line Argument	Interpretation
<code>-all</code>	Generates all starting point code: types, service proxy, service interface, server mainline, client mainline, implementation object, and an Ant <code>build.xml</code> file.
<code>-ant</code>	Generates the Ant <code>build.xml</code> file.
<code>-nexclude</code> <code><schema-namespace[=java-package-name]></code>	Ignore the specified WSDL schema namespace when generating code. This option may be specified multiple times. Also, optionally specifies the Java package name used by types described in the excluded namespace(s).
<code>-exsh <(true/false)></code>	Enables or disables processing of extended soap header message binding.
<code>-validate</code>	Enables or disables validating the wsdl before generate the code
<code>-v</code>	Displays the version number for the tool.
<code>-verbose</code>	Displays comments during the code generation process.
<code>-quiet</code>	Suppresses comments during the code generation process.
<code><wsdlfile></code>	The path and name of the WSDL file to use in generating the code.

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 uses a Web service endpoint's implementation (SEI) class and associated types classes to generate a WSDL file. The syntax of the command is summarized in the following illustration.

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

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

Command Line Argument	Interpretation
<code>-?</code> <code>-help</code> <code>-h</code>	Displays the online help for this utility.
<code>-o <output-file></code>	Specifies the name of the generated WSDL file.
<code>-cp <class-path></code>	Specify the SEI and types class search path of directories and zip/jar files.
<code>-t <target-namespace></code>	Specifies the target namespace to use in the generated WSDL file.

Command Line Argument	Interpretation
<code>-servicename <service-name></code>	Specifies the value of the generated <code><service></code> element's <code>name</code> attribute.
<code>-v</code>	Displays the version number for the tool.
<code>-verbose</code>	Displays comments during the code generation process.
<code>-quiet</code>	Suppresses comments during the code generation process.
<code><classname></code>	Specifies the name of the service endpoint class.

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 classes. Make certain these classes are on the `CLASSPATH` or in a location identified through the `-cp` flag.

The Celtix wsdl2soap Utility

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

```
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>
```

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

Command Line Argument	Interpretation
<code>-?</code> <code>-help</code> <code>-h</code>	Displays the online help for this utility.
<code>-i <port-type-name></code>	Specifies the <code>portType</code> element for which a binding should be generated.
<code>-b <binding-name></code>	Specifies the name of the generated SOAP binding.
<code>-d <output-directory></code>	Specifies the directory to place generated WSDL file.
<code>-o <output-file></code>	Specifies the name of the generated WSDL file.
<code>-n <soap-body-namespace></code>	Specifies the SOAP body namespace when the style is RPC.
<code>-style <(document/rpc)></code>	Specifies the encoding style (<code>document</code> or <code>RPC</code>) to use in the SOAP binding. The default is <code>document</code> .
<code>-use <(literal/encoded)></code>	Specifies the binding use (<code>encoded</code> or <code>literal</code>) to use in the SOAP binding. The default is <code>literal</code> .
<code>-v</code>	Displays the version number for the tool.

The Celtix wsdl2soap Utility: The Celtix wsdl2soap Utility

Command Line Argument	Interpretation
<code>-verbose</code>	Displays comments during the code generation process.
<code>-quiet</code>	Suppresses comments during the code generation process.
<code><wsdlurl></code>	The path and name of the WSDL file containing the <code>portType</code> element definition.

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

The Celtix wsdl2service Utility

The `wsdl2service` tool creates a new WSDL file containing an HTTP or JMS service definition from an existing WSDL document containing a `binding` element. The syntax of the command is summarized in the following illustration.

```
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>
```

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

Command Line Argument	Interpretation
<code>-?</code> <code>-help</code> <code>-h</code>	Displays the online help for this utility.
<code>-transport <http/jms></code>	Specifies the type of transport to use for the generated service.
<code>-e <service-name></code>	Specifies the value of the generated <code>service</code> element's <code>name</code> attribute.
<code>-p <port-name></code>	Specifies the value of the generated <code>port</code> element's <code>name</code> attribute. To specify multiple <code>port</code> elements, separate the names by a space.
<code><address></code>	Specifies the value used in the <code>address</code> element of the port.
<code>-n <attribute-binding-name></code>	Specifies the binding used to generate the service.
<code>-o <output-file></code>	Specifies the name of the generated WSDL file.
<code>-d <output-directory></code>	Specifies the directory in which the generated WSDL is placed.

Command Line Argument	Interpretation
<code>-v</code>	Displays the version number for the tool.
<code>-verbose</code>	Displays comments during the code generation process.
<code>-quiet</code>	Suppresses comments during the code generation process.
<code><wsdlurl></code>	The path and name of the existing WSDL file.

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 containing a `<portType>` definition. The syntax of the command is summarized in the following illustration.

```
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>
```

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

Command Line Argument	Interpretation
<code>-?</code> <code>-help</code> <code>-h</code>	Displays the online help for this utility.
<code>-i <port-type-name></code>	Specifies the <code>portType</code> element to use.
<code>-b <binding-name></code>	Specifies the name of the generated XML binding.
<code>-e <service-name></code>	Specifies the value of the generated <code><service></code> element's <code>name</code> attribute.
<code>-p <port-name></code>	Specifies the value of the generated <code><port></code> element's <code>name</code> attribute. To specify multiple <code><port></code> elements, separate the names by a space.
<code>-a <address></code>	Specifies the value used in the <code><address></code> element of the generated <code><port></code> element.
<code>-d <output-directory></code>	Specifies the directory to place generated WSDL file.
<code>-o <output-file></code>	Specifies the name of the generated WSDL file.
<code>-v</code>	Displays the version number for the tool.
<code>-verbose</code>	Displays comments during the code generation process.

The Celtix wsdl2xml Utility: The Celtix wsdl2xml Utility

Command Line Argument	Interpretation
<code>-quiet</code>	Suppresses comments during the code generation process.
<code><wsdlurl></code>	The path and name of the existing WSDL file.

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 file containing a `<types>` element populated by the types defined in the XMLSchema document. The syntax of the command is summarized in the following illustration.

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

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

Command Line Argument	Interpretation
<code>-?</code> <code>-help</code> <code>-h</code>	Displays the online help for this utility.
<code>-t <target-name-space></code>	Specifies the target namespace for the generated WSDL.
<code>-n <wsdl-name></code>	Specifies the value of the generated <code><definition></code> element's <code>name</code> attribute.
<code>-d <output-directory></code>	Specifies the directory in which the generated WSDL is placed.
<code>-o <output-file></code>	Specifies the name of the generated WSDL file.
<code>-v</code>	Displays the version number for the tool.
<code>-verbose</code>	Displays comments during the code generation process.
<code>-quiet</code>	Suppresses comments during the code generation process.
<code><xsdurl></code>	The path and name of the existing XSDSchema file.

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

The Celtix wsdlvalidator Utility

The `wsdlvalidator` tool validate whether a wsdl file or wsdl url is well-formed ,schema confromed.

```
wsdlvalidator -d <schema-directory> -s <schema-url>*
              -h -v -verbose -quiet <wsdlurl>
```

The arguments used to validate WSDL file are reviewed in the following table.

Command Line Argument	Interpretation
-? -help -h	Displays the online help for this utility.
-d <schema-directory>	Specifies the directory used to search for schemas
-s <schema-url>	Specifies the URL of a user specific schema to be included in the validation of the contract. This switch can appear multiple times.
-v	Displays the version number for the tool.
-verbose	Displays comments during the validation.
-quiet	Suppresses comments during the validation
<wsdlurl>	The path and name of the existing WSDL file