

Frequently Asked Questions

1 Questions

1.1 1. What is the official version of WSIF? How to I get the latest version?

The official version of WSIF is found here on the Apache web site. The wsif-src*.zip contains source distribution, the wsif-bin* zip file contains a built wsif.jar along with the API Javadoc and the wsif-all*.zip contains the source, the JAR and the javadocs. (* will be replaced by the current version number in the file names). The version available from Alphaworks is no longer maintained.

1.2 2. What are the differences between WSIF and WSTK?

WSIF is a framework for invoking WSDL-described services. WSTK is IBM's Web Services Toolkit, a preview technology for web services support in Websphere available from alphaworks. WSTK includes WSIF amongst other things.

1.3 3. What are the differences between WSIF and Axis?

Axis is an implementation of SOAP. It includes on the server-side infrastructure for deploying web service implementations and then routing SOAP messages between clients and those implementations. It also implements the JAX-RPC specification for invoking SOAP services. WSIF is similar to the client piece of Axis, in that it is used for invoking services. However, WSIF's API is WSDL-driven and protocol independent; it allows protocol-specific code ("providers") to be plugged in. For invoking SOAP services, WSIF is in fact packaged with an Axis provider, that uses Axis APIs (i.e. JAX-RPC) to do the invocation. So WSIF operates at a more abstract level than Axis.

1.4 4. What are the differences between WSIF and JAX-RPC?

JAX-RPC is an API for invoking XML-based RPC services - essentially its current scope is limited to invocation of SOAP services. WSIF is an API for invoking WSDL-described services, whether they happen to be SOAP services or not (for example, WSIF defines WSDL bindings so that EJBs, enterprise software accessible using JMS or the Java Connector architecture as well as local java classes can all be described as first class WSDL services and then invoked using the same, protocol-independent WSIF API)..

1.5 5. How do I set an HTTP proxy to be used by WSIF?

HTTP proxy settings can be set using the following system properties
http.proxyHost - The hostname of the proxy server
http.proxyPort - The port for the proxy server
http.nonProxyHosts - A comma separated list of hosts to access directly rather than through the proxy

1.6 6. How can I configure WSIF to work through an authenticating proxy server?

The following applies to the latest nightly builds and is not available in WSIF 2.0 In order to retrieve and parse a wsdl document from behind an authenticating proxy, you can use the WSIFUtils.readWSDLThroughAuthProxy method. Assuming a username of fred and password of cat then use: String wsdlLoc = "http://someserver.com/somedoc.wsdl" char[] passwd = "cat".toCharArray(); java.net.PasswordAuthentication pa = new java.net.PasswordAuthentication("fred", c); Definition def = WSIFUtils.readWSDLThroughAuthProxy(wsdlLoc, pa); If using the Axis provider, you can invoke a service through an authenticating proxy by setting proxy usernames and passwords in one of two ways: Set system properties http.proxyUser and http.proxyPassword or Set the username and password on the context message used by the service via: WSIFService service = // Get from factory WSIFMessage ctx = service.getContext(); ctx.setObjectPart(WSIFConstants.CONTEXT_HTTP_PROXY_USER, "fred"); ctx.setObjectPart(WSIFConstants.CONTEXT_HTTP_PROXY_PWSD, "cat"); service.setContext(ctx);

1.7 7. Where can I find details of the extensions to WSDL that are supported by WSIF?

Links to the supported extensions can be found in the providers section.

1.8 8. How do I define my own WSDL binding extensions and write WSIF providers for them

Here's how to write your own WSDL extensions to support new protocols, and then write your own WSIF providers to support customized WSIF extensions.

1.9 9. How do I report problems and request new features?

The mailing lists are what you are looking for

1.10 10. How do I select SOAP provider used in WSIF?

WSIF comes with an Apache Axis provider so using axis with WSIF is easy. WSIF has two SOAP providers, one using Apache SOAP 2.3 and the other using Axis, by default WSIF will use the Axis provider. The default is controlled by the org.apache.wsif.util.WSIFPluggableProviders class. The javadoc for this class gives

Frequently Asked Questions

details of how the default is defined, but it can be overridden programmatically, for example, the following call sets the axis provider to be the default: `WSIFPluggableProviders.overrideDefaultProvider("http://schemas.xmlsoap.org/wsdl/soap/", new WSIFDynamicProvider_ApacheSOAP());`

1.11 11. How do I use use dynamic invoker sample?

```
java clients.DynamicInvoker http://www.xmethods.net/sd/2001/TemperatureService.wsdl
getTemp 10570 Reading WSDL document from
'http://www.xmethods.net/sd/2001/TemperatureService.wsdl' Preparing WSIF dynamic
invocation - WSIF0006W: Multiple WSIFProvider found supporting the same namespace
URI 'http://schemas.xmlsoap.org/wsdl/soap/'. Found
('org.apache.wsif.providers.soap.apachesoap.WSIFDynamicProvider_ApacheSOAP,
org.apache.wsif.providers.soap.apacheaxis.WSIFDynamicProvider_ApacheAxis') -
WSIF0007I: Using WSIFProvider
'org.apache.wsif.providers.soap.apachesoap.WSIFDynamicProvider_ApacheSOAP' for
namespaceURI 'http://schemas.xmlsoap.org/wsdl/soap/' Executing operation getTemp Result:
return=56.0 Done!
```

1.12 12. Sample DynamicInvoker and Complex Type handling in WSIF?

WSIF does support complex types - of course XML schema support is limited (but very reasonable). The DynamicInvoker doesn't support invocation of services using complex types since this requires that java representations of the complex types be generated first. It is also possible to extend DynamicInvoker to support complex types but it will require some kind of intermediate representation of values that can handle any complex type (such as JROM...)