# Web Services Invocation Framework: Quick Start

## 1 Preliminaries

Download the WSIF distribution from http://cvs.apache.org/dist/axis/wsif/. It comes in three flavors: the binary distribution ("bin"), the source-only distribution ("src") or everything included ("all"). If you are familiar with CVS and want to work with the latest code use cvs. Using CVS you can also retrieve any past release by using tag (for example WSIF_2_0_ALPHA2).

## 2 Getting Started

If you have downloaded binary package then you are ready to use WSIF: the very first thing to do is to check provided samples. However before you start please read WSIF FAQ. Then read about how to run the samples. Otherwise if you downloaded source code or accessed it from CVS you will need to build WSIF. To do this, first check that all prerequisites are available. Then proceed with building.

After you have successfully downloaded and installed WSIF, you can test your installation by running the samples. If you are able to run all the samples successfully you should have a very good idea of what you can do with WSIF. Essentially once you have a WSDL file with a binding that WSIF understands (such as SOAP, EJB, Java, JMS, JCA, etc.) you can write a client that uses WSIF's APIs - the dynamic invocation API or the stub-based invocation - to use this service.

Traditionally, the burden of dealing with new protocols has been a carried by the server-side. Software has to morph itself (through the addition of wrappers) to look like something the client expects to see. With WSIF, the software remains exactly the same - we just get the client to use WSIF's binding independent API, and through the addition of providers, we can mix and match protocols as we please.

*Service* is a very liberal term for WSIF! Anything that can be described in WSDL qualifies as a service and is something you can access using the WSIF API. Moreover, since WSDL is extensible, potentially everything can be described using WSDL. The next section describes how you can define your own bindings and write your own WSIF providers.