

# WebServices - SOAP

## 1. WebServices - SOAP - General Features

Supports most of the SOAP v1.1 specification. Supports the SOAP Messages with Attachments W3C Note. Provides server-side infrastructure for deploying, managing and running SOAP enabled services. Provides client-side API for invoking SOAP services. Release includes full source under the Apache Software License. Supports three encoding styles: SOAP v1.1 Encoding, Literal XML and XMI. XMI encoding (available when using Java 1.2.2) supports automatic marshalling and unmarshalling of arbitrary objects. SOAP encoding: built-in support is provided for encoding/decoding primitive types, Strings, arbitrary JavaBeans (using reflection) and 1-dimensional arrays of these types. For other types user can hand-write encoder/decoder and register with XML-SOAP runtime. Literal XML encoding: allows one to send XML elements (DOM org.w3c.dom.Element objects) as parameters by embedding the literal XML serialization of the DOM tree. No code needs to be written to support this (see the addressbook demo to see a sample use of it). Supports messaging and RPC over two transports: HTTP and SMTP. Supports authoring services in scripting languages.

## 2. WebServices - SOAP - Implementation Restrictions

The following features of the SOAP v1.1 specification are not currently supported: encodingStyle attribute must have only one encoding style given (see section 4.1.1 of the spec) mustUnderstand attribute support - only supports checking for and rejecting requests that require mustUnderstand checking root attribute actor attribute and SOAP intermediaries does not use multi-ref accessors during serialization The following limitations on SOAP Messages with Attachments currently exist: The document base URI is not picked up from the multipart's Content-Location header. Support for relative URIs in Content-Location headers is limited to concatenating the document base URI to the relative URI. The provided SMTP transport does not support multipart messages. Server-side RPC methods have no way to add attachments to the response other than via the return object. Messaging methods can do this already.