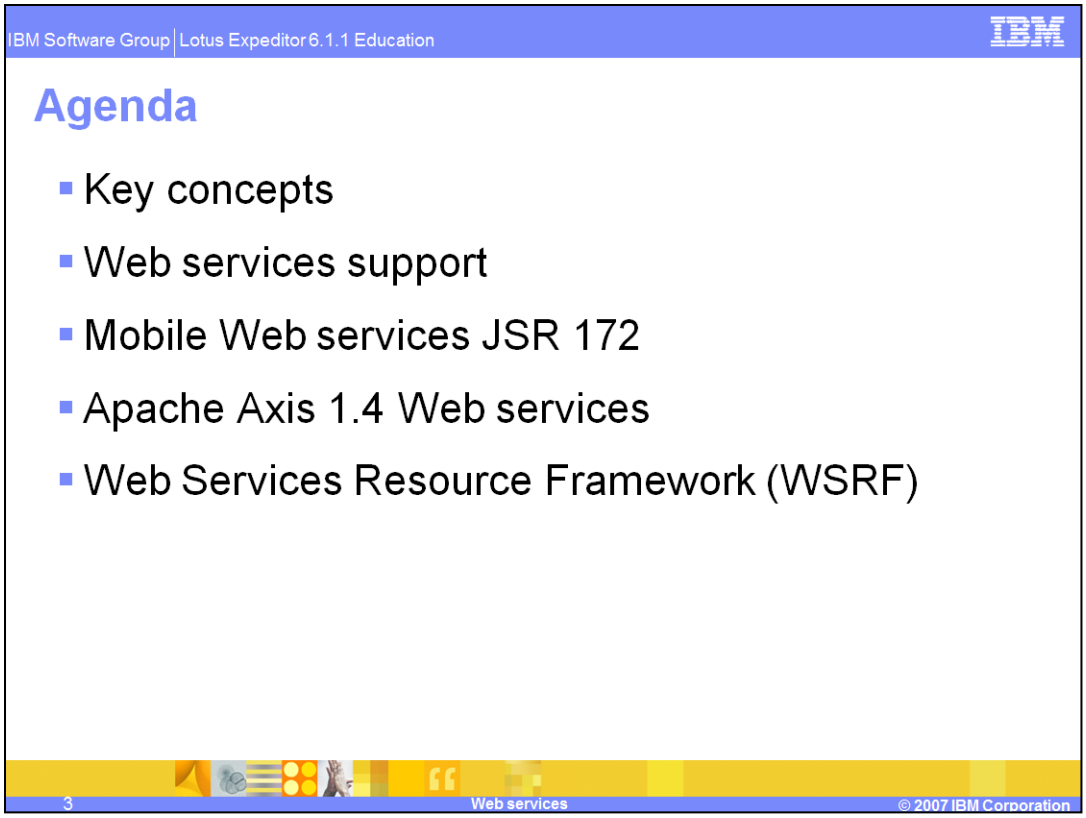
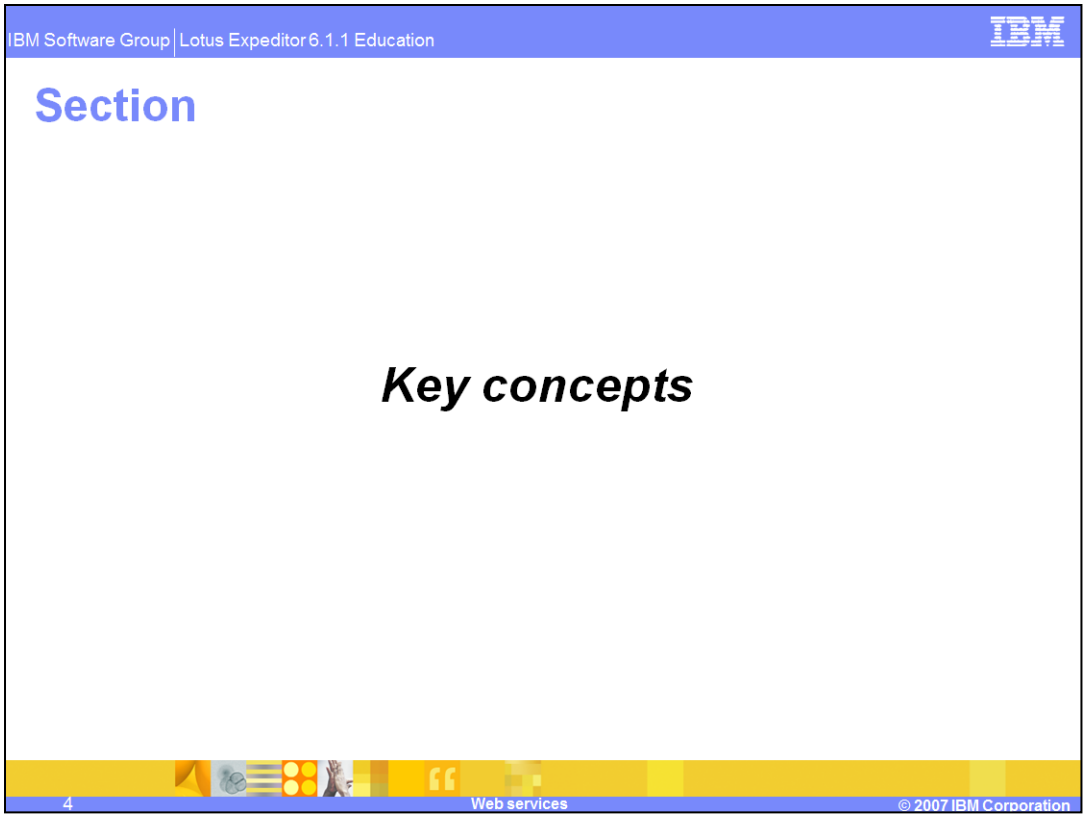
This presentation explains the Web Services capabilities in IBM Lotus Expeditor Client for Desktop.

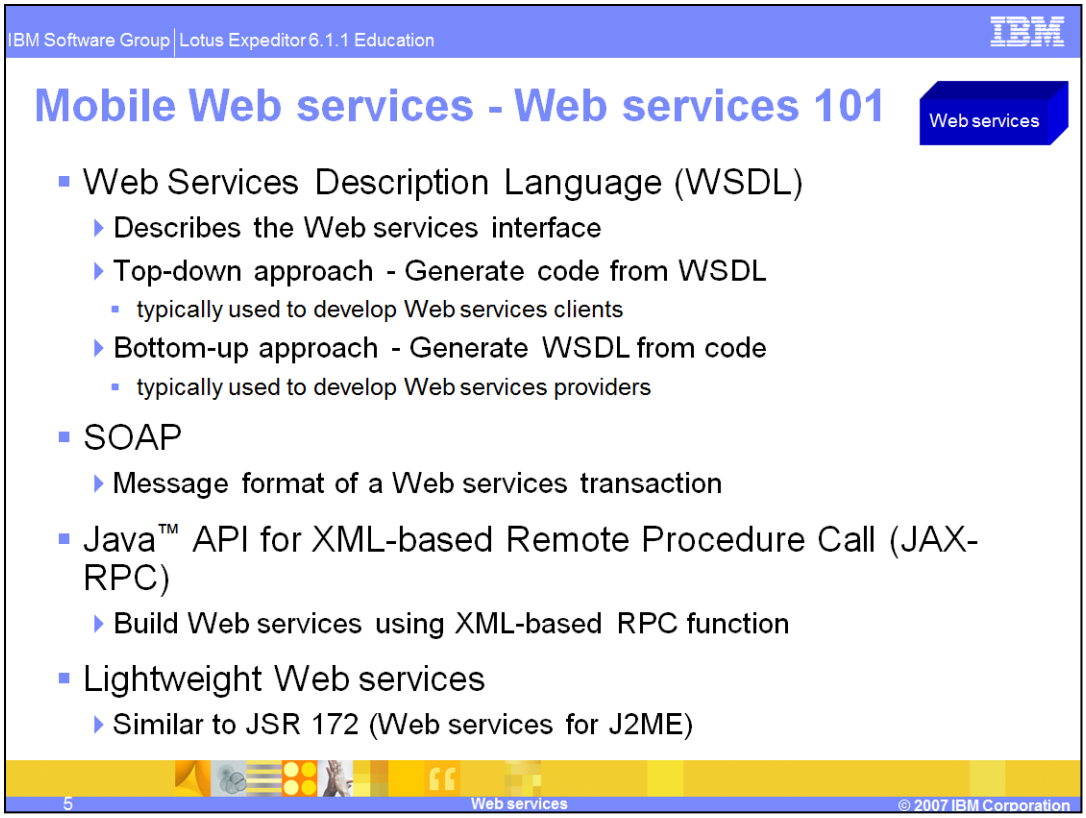
The goal of this presentation is to understand the Web Services support provided by IBM Lotus Expeditor Client for Desktop.



The agenda of this presentation is to explain key concepts and describe the Web Services support provided by the client.



This presentation will start with an overview of key Web Services concepts.



Mobile Web Services enable you to develop applications that consume and provide Web services. Before explaining the details of mobile Web services, this presentation will review some basic concepts.

A Web Services Description Language (or WSDL) document provides the description of the Web services interface. Web services can be created using a top-down or bottom-up approach. A top-down approach is used to generate code from a WSDL (typically used for developing Web services clients), whereas a bottom-up approach is used to generate a WSDL from code (typically used for developing Web Services providers). However, the IBM WebSphere® Everyplace Client Toolkit Version 6 Web Services plug-in currently supports only the top-down approach. For more information about WSDL, visit the URL http://www.w3.org/TR/wsdl.

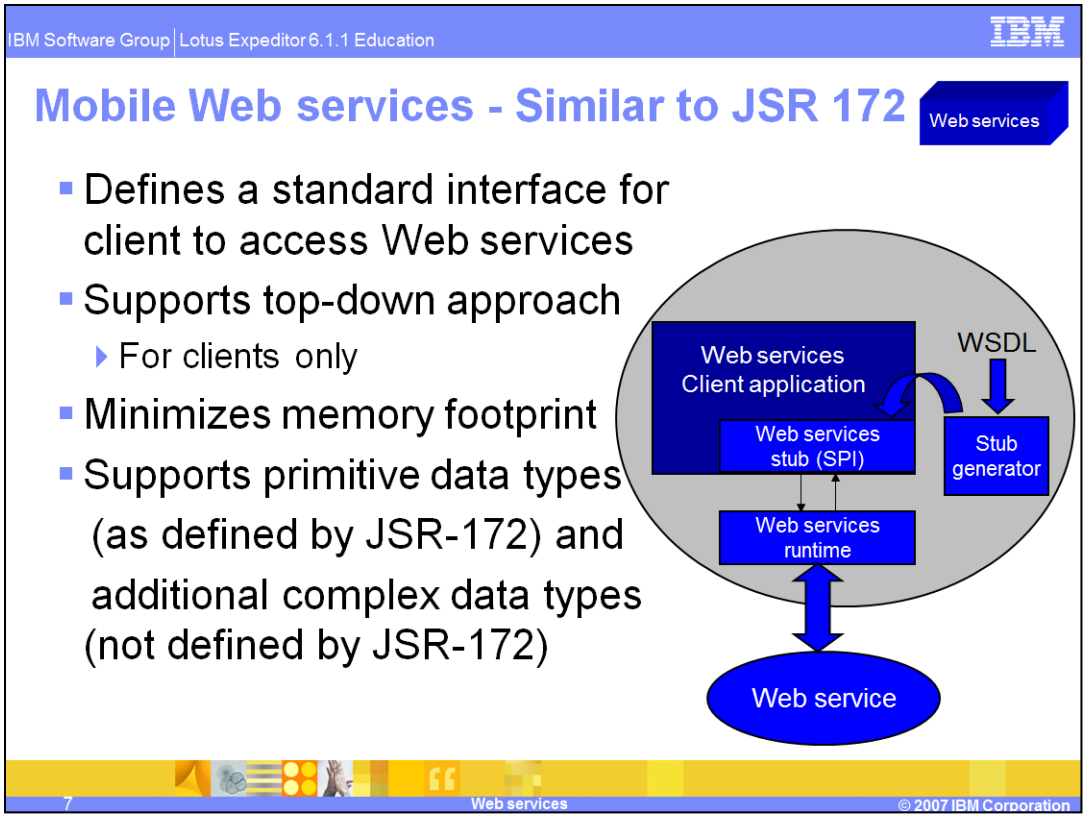
SOAP is the message format of the transaction that takes place when a Web Services client communicates with a Web Services provider. The WSDL defines the restrictions on the format of these messages. For more information about SOAP, see http://www.w3.org/TR/soap.

JAX-RPC The Java API for XML-Based Remote Procedure Call (JAX-RPC) enables developers to build Web Services using XML-based RPC functionality according to the SOAP 1.1 specification. For more information about JAX-RPC, see http://java.sun.com/xml/jaxrpc.

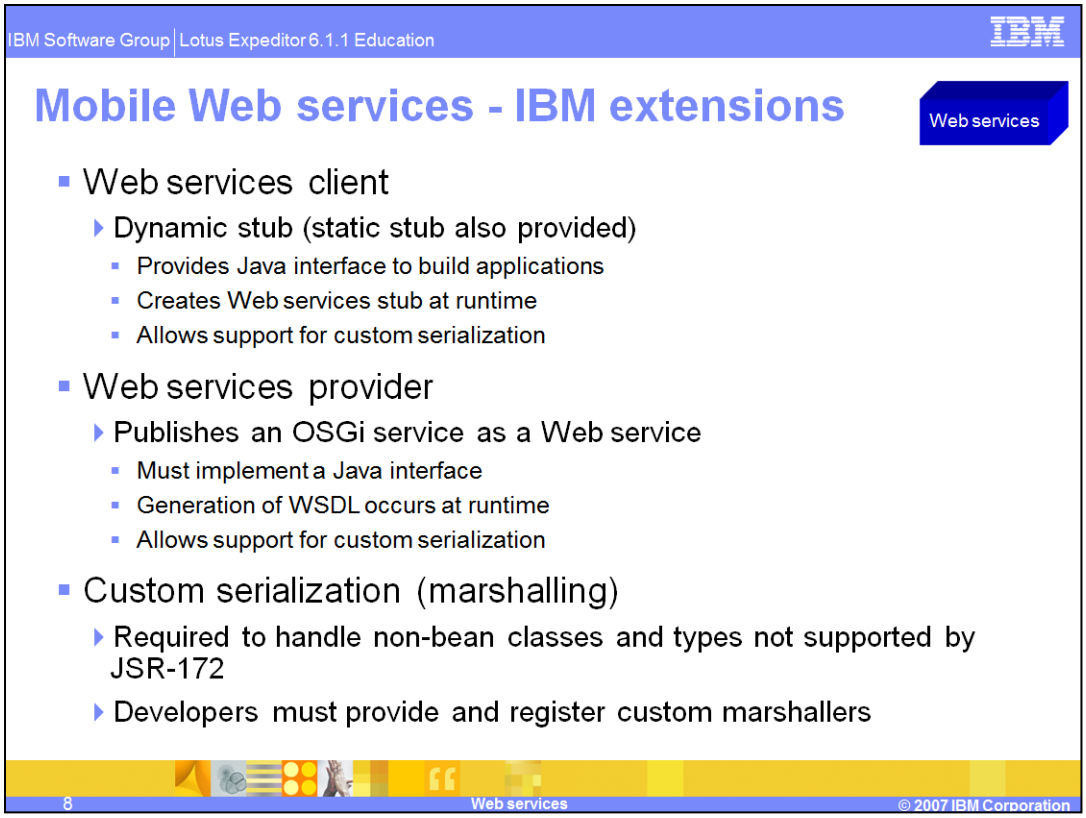
Mobile Web Services is a light weight implementation that provides functionality similar to

libraries that implement the Java 2 Micro Edition Web Services Specification (JSR-172).

Next, let's explore the Web Services support provided by the client platform.



JSR 172 defines a standard interface for a client application to access Web Services. Consistent with JSR 172, the Mobile Web Services Client wizard generates a static client stub class using the WSDL that is exported from the Web Services provider as its input. The stub is then used by a Web Services client application to invoke the Web Services provider. JSR 172 specifies support for primitive and complex data types, for example, Boolean, byte, short, int, long, float, double, String, complex types (a type that allows elements in its content and may carry attributes), and arrays of primitive and complex types.



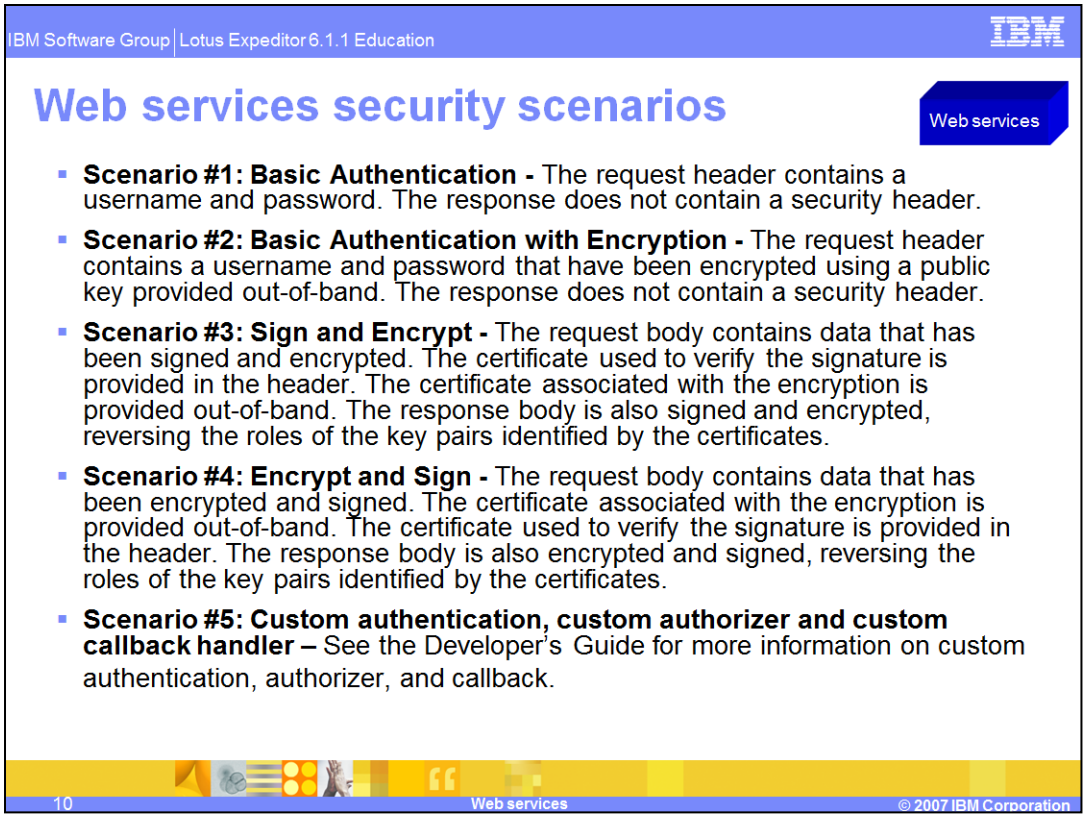
A dynamic stub allows you to decide to use Web services at runtime rather than build time, that is, dynamic stub allows Web services to be configured and deployed in the field without involving a build team. A dynamic stub also allows a Web services client to create and use custom marshallers for WSDL types that are non-bean classes or incompatible with JSR-172.

For a Web services provider, any OSGi service can be exposed as a Web services provider using the toolkit, provided that the service implements a Java interface. Generation of a WSDL-document occurs at runtime using Java reflection into the OSGi service class.

For custom serialization, if Web services need to handle non-bean classes or types that are incompatible with JSR-172, then you can provide and register custom marshallers to handle these classes or types.

Web Services security is based on the WS-Security Minimalist Profile specification from OASIS (or Organization for the Advancement of Structured Information Standards), which is used to secure SOAP messages. Web Services protects messages through support of 4 key OASIS Web Services scenarios, which will be explained on the next slide.

Web Services security works with WebSphere Application Server 5.1 and 6.0.



This slide explains the four Web Services Security scenarios from OASIS that are supported by Mobile Web Services.

SSL support

Web services

- Support on the client side
  - You can specify these properties for the client:

    -Djavax.net.ssl.keyStore=<path_to_keystore_file>

    -Djavax.net.ssl.keyStoreType=<keystore_type>

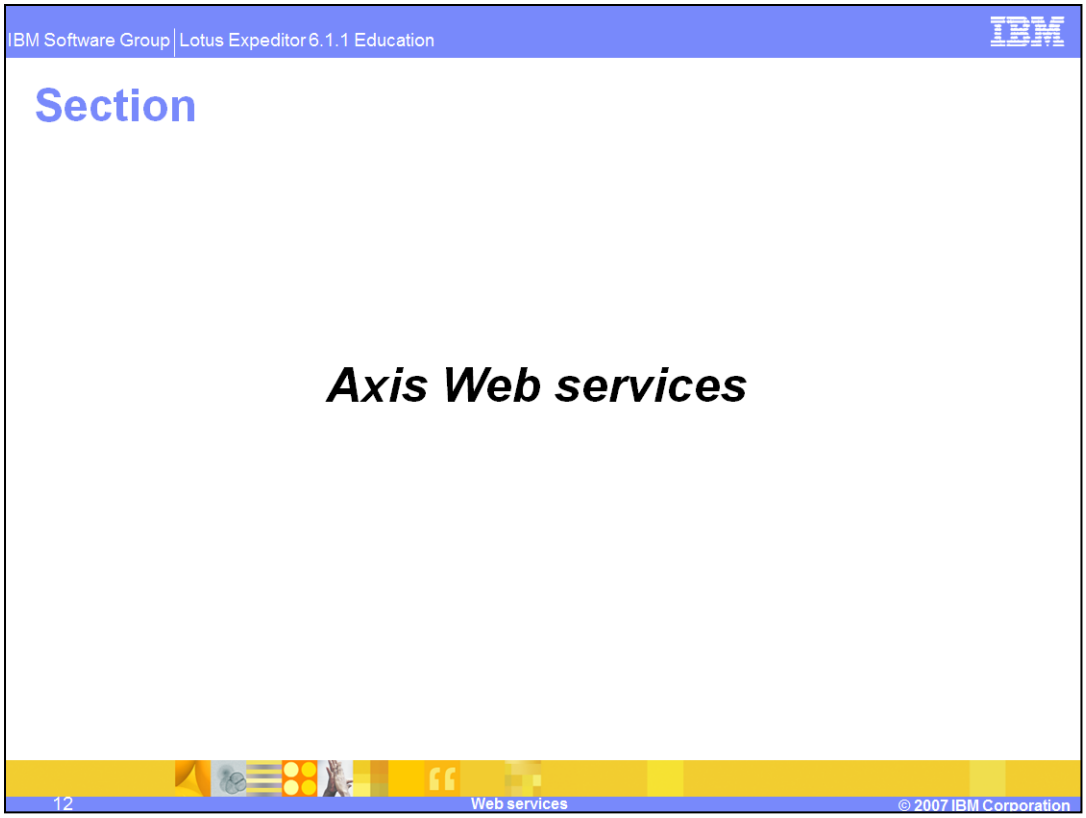
    -Djavax.net.ssl.keyStorePassword=<keystore_password>

    -Djavax.net.ssl.trustStore=<path_to_truststore_file>

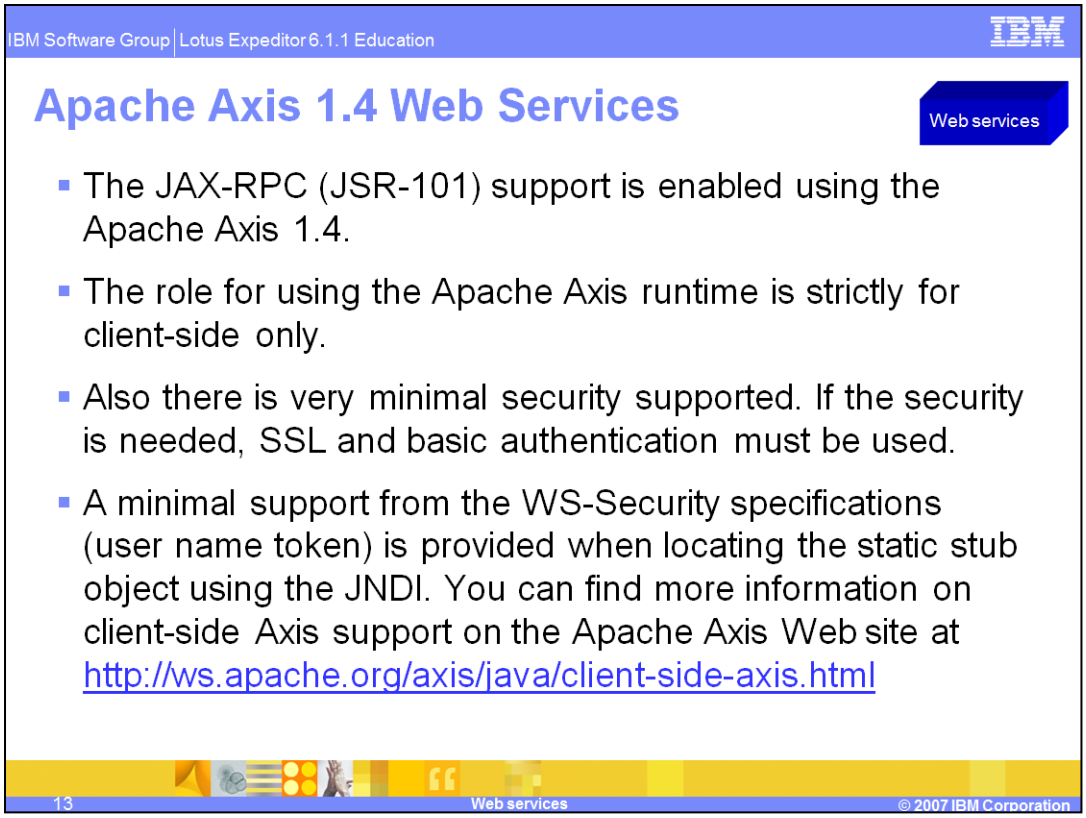
    -Djavax.net.ssl.trustStoreType=<truststore_type>

    -Djavax.net.ssl.trustStorePassword=<truststore_password>
  - Specified in the rpcinstall.properties file
  - See System Administration Guide or Developer's Guide for additional information
- Support on the provider side
  - SSL connection managed by Web container
  - Configure the Web container

This slide explains how SSL support is available for Web Services clients and providers.



Next, let's explore the Web Services support provided by the client platform.



The JAX-RPC (JSR-101) support is enabled using the Apache Axis 1.4.

The role for using the Apache Axis runtime is strictly for Client-Side only.

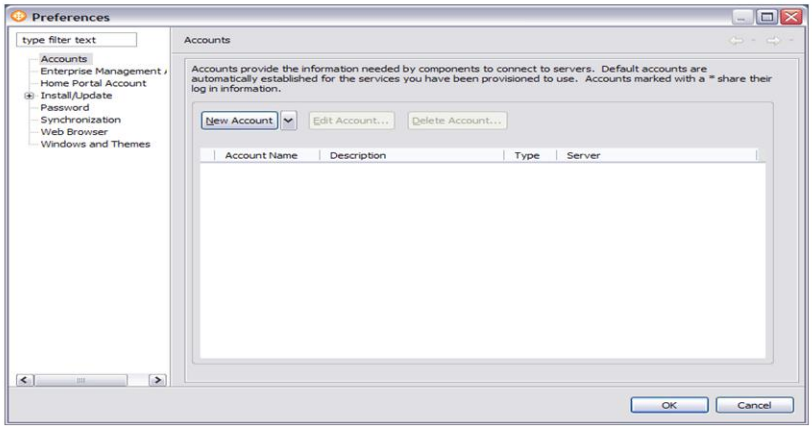
Also there is very minimal security supported. If the security is needed, SSL & Basic Authentication must be used.

A minimal support from the WS-Security specifications (User name Token) is provided when locating the static stub object using the JNDI. You can find more information on Client-Side Axis support on the Apache Axis Web site at the address on this slide.

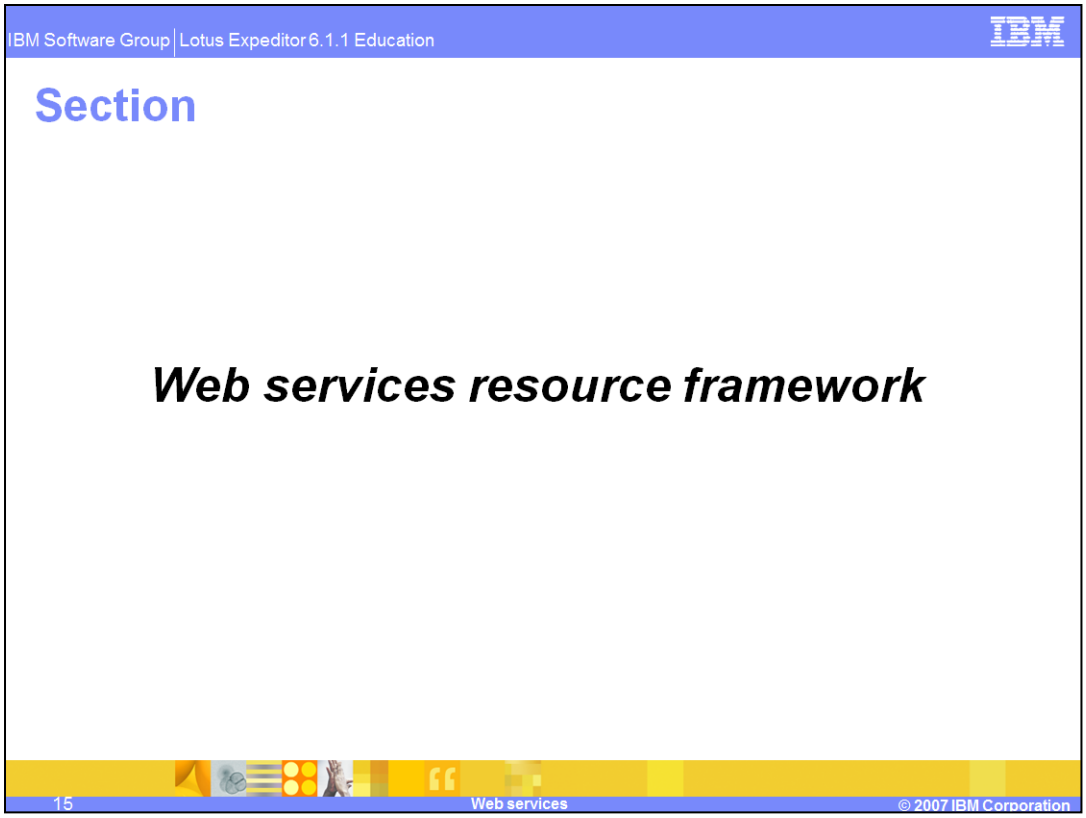
Web Services – Accounts and Axis

Web services

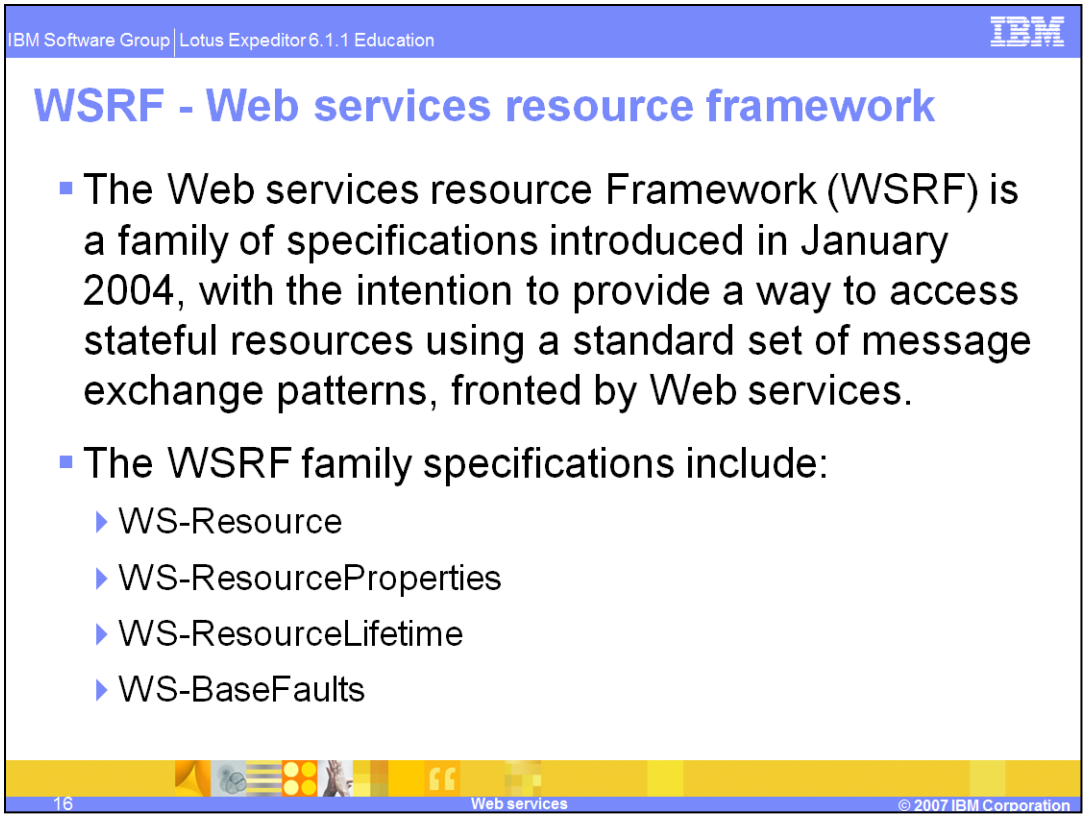
- Accounts can be created using the Accounts Preferences page.



You can create accounts for Apache Axis Web Services clients, using the Account Preference page.



Next let's cover key concepts of Web Services Resource Framework (or WSRF).



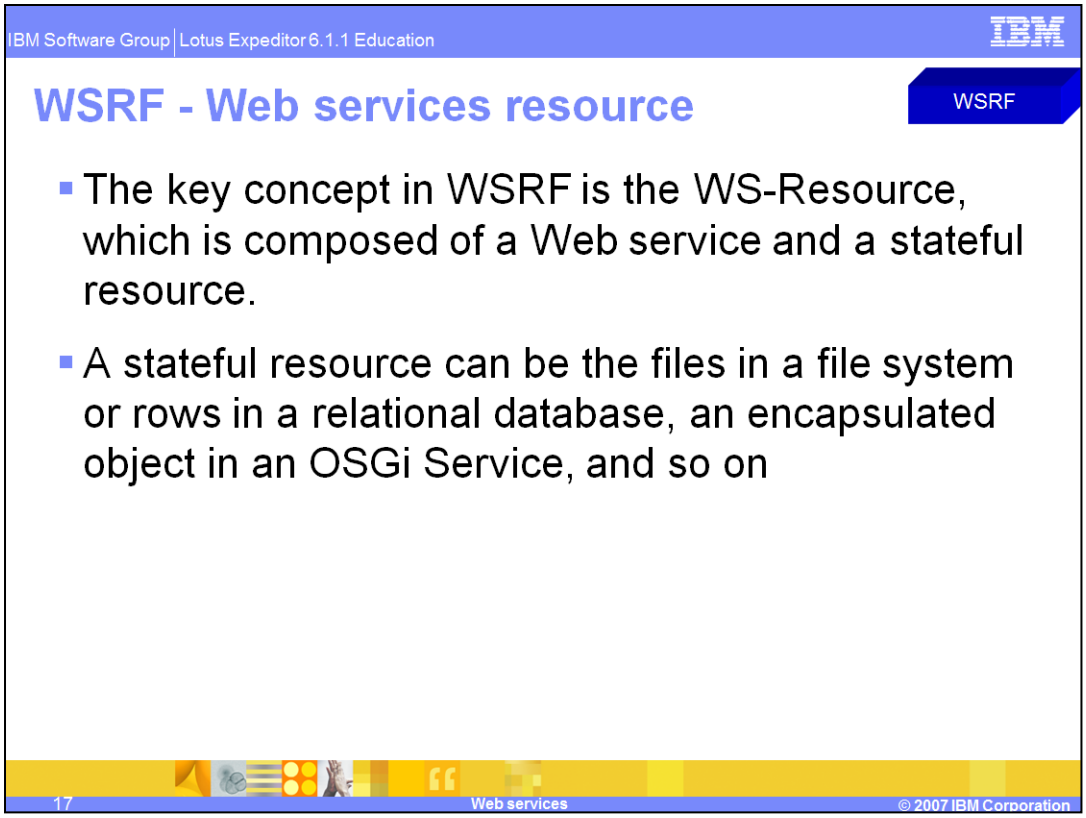
The Web services resource framework is a family of specifications introduced in January 2004, with the intention to provide a way to access stateful resources using a standard set of message exchange patterns, fronted by Web services. The WSRF family specifications include:

WS-Resource

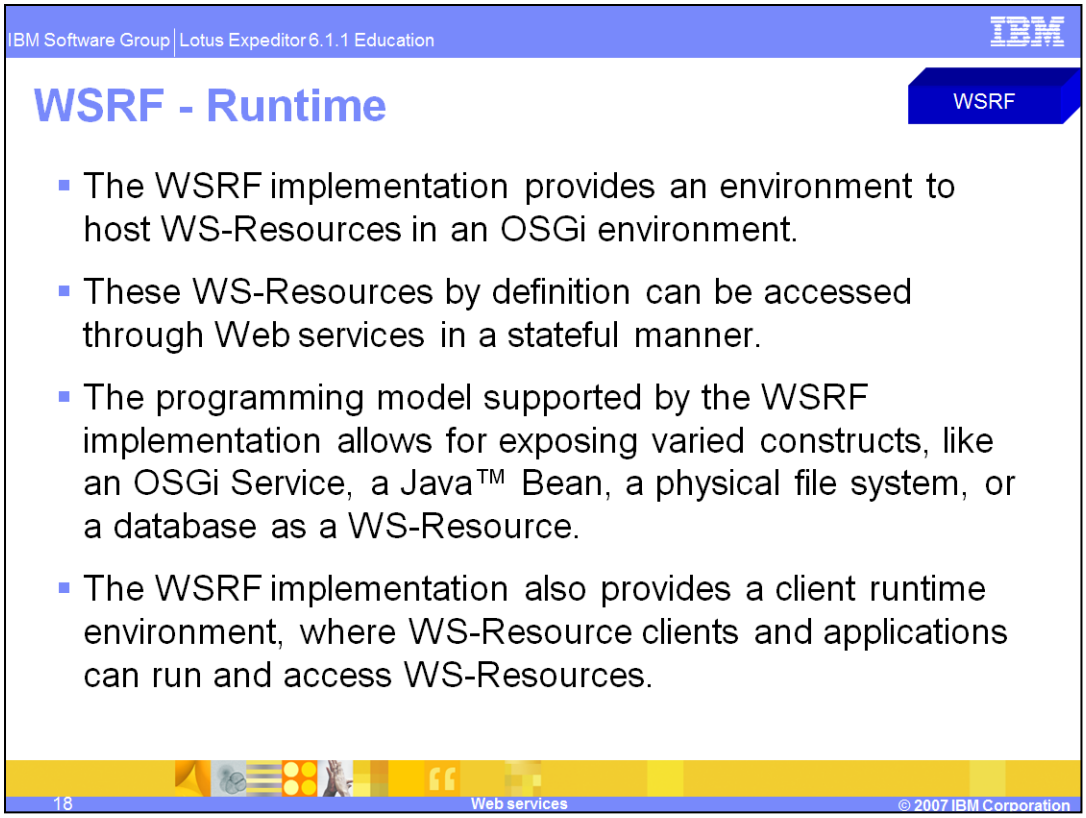
WS-ResourceProperties

WS-ResourceLifetime

WS-BaseFaults

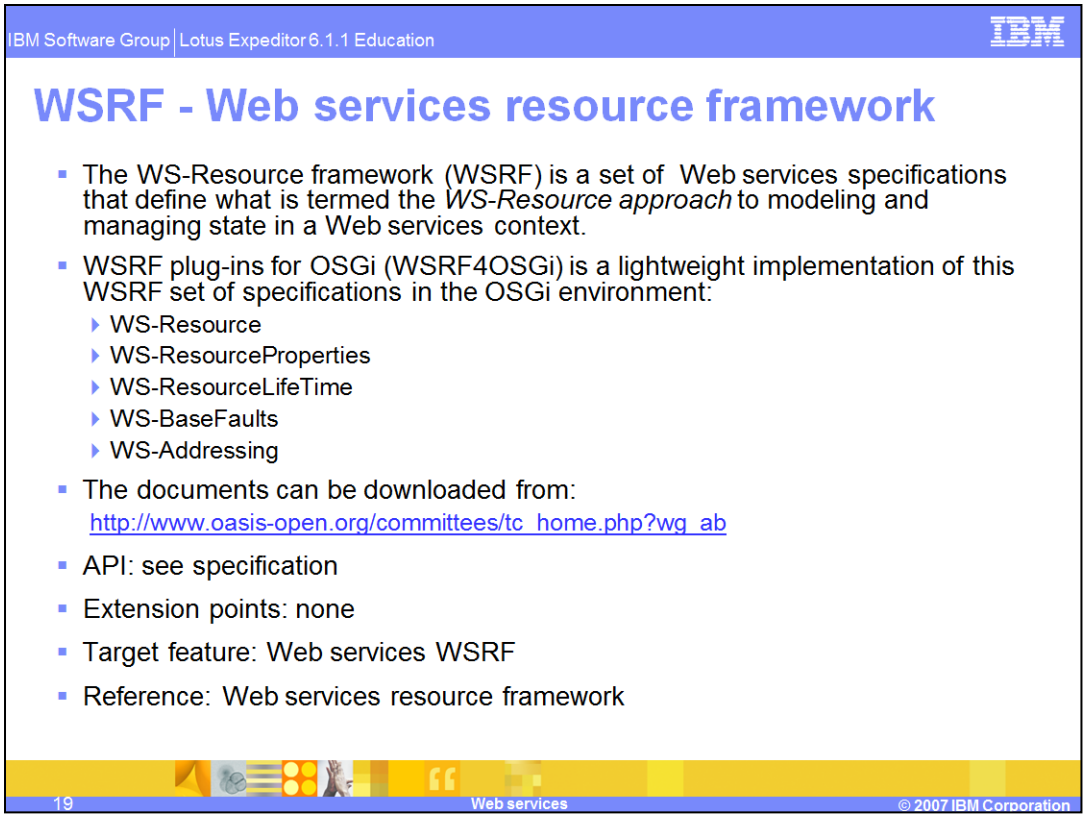


The key concept in WSRF is the WS-Resource, which is composed of a Web service and a stateful resource. A stateful resource can be the files in a file-system or rows in a relational database, or an encapsulated object in an OSGi Service.



The WSRF implementation provides an environment to host WS-Resources in an OSGi environment. These WS-Resources by definition can be accessed through Web services in a stateful manner. The programming model supported by the WSRF implementation allows for exposing varied constructs, like an OSGi service, a Java bean, a physical file system, or a database as a WS-Resource.

The WSRF implementation also provides a client runtime environment, where WS-Resource clients and applications can run and access WS-Resources.



The WSRF component of Web services runtime extends the applicability of the Web in other application domains like system management, and autonomic computing.

The Web Services Resource Framework defines a family of specifications for accessing stateful resources using Web services.

The WS-Resource Framework (WSRF) is a set of six Web services specifications that define what is termed the *WS-Resource approach* to modeling and managing state in a Web services context.

It is a set of specifications that include:

- WS-Resource
- WS-ResourceProperties
- WS-ResourceLifeTime
- WS-BaseFaults
- WS-Renewable References
- WS-ServiceGroup.

These WSRF specifications are currently not supported by WSRF4OSGi implementation:

- WS-Renewable References

•WS-Service Group

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM Lotus WebSphere

J2ME, Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.