Sample Projects

Apache CXF Example Projects

The following tables provide information about a subset of the example projects provided by Apache CXF in the standard distributions.

JAX-WS Examples

Example Name	Example Description	Example Location
Java First using JAX-WS APIs	Sample illustrates how to develop a service using the "code first" approach with the JAX-WS APIs.	java_first_jax ws
Java First demo service using the JAXWSFactoryBeans	Sample illustrates how to develop a service using the JAXWSFactoryBeans. It also makes use of LoggingInterceptors. The sample takes the "code first" approach using JAX-WS APIs.	java_first_jax ws_factory_b ean
Java First POJO Sample	Sample illustrates how to develop a service that is "code first", POJO-based. This sample uses the JAXB Data binding by default, but you can use Aegis Data binding by removing a few lines detailed in the README.txt file.	java_first_pojo
Spring HTTP Sample	Sample will lead you through creating your first service with Spring. You'll learn how to write a simple JAX-WS "code-first" service, set up the HTTP Servlet transport and use CXF's Spring beans.	java_first_sp ring_support
JAX-WS Asynchronous Demo using Document /Literal Style	Sample using Document/Literal Style sample illustrates the use of the JAX-WS asynchronous invocation model.	jaxws_async
JAX-WS Dispatch/Provider Demo	Sample demonstrates the use of JAX-WS Dispatch and Provider interface.	jaxws_dispat ch_provider
JAX-WS Handler Demo	Sample shows how JAX-WS handlers are used. The server uses a SOAP protocol handler which logs incoming and outgoing messages to the console.	jax_ws_hand lers
RESTful Hello World Demo	Sample shows REST based Web Services using the JAX-WS Provider/Dispatch.	restful_dispa tch

WSDL-First Examples

Example Name	Example Description	Example Location
WSDL First	Sample shows how to build and call a web service using a given WSDL (also called Contract First).	wsdl_first
CXF Dynamic Client Sample	Sample illustrates the use of the CXF dynamic client against a standalone server using SOAP 1.1 over HTTP.	wsdl_first_dyn amic_client
WSDL first demo using HTTPS	Sample takes the hello world sample a step further by doing the communication using HTTPS.	wsdl_first_https
WSDL first demo using BARE Style in XML Binding (pure XML over HTTP)	Sample illustrates the use of Apache CXF's xml binding. This specific sample shows you how xml binding works with the doc-lit bare style.	wsdl_first_pur e_xml
WSDL first demo using RPC-Literal Style	Sample demonstrates the use of the hello world sample with RPC-Literal style binding.	wsdl_first_rpclit
WSDL first demo using SOAP12 in Document /Literal Style	Sample shows the use of Apache CXF's SOAP 1.2 capabilities.	wsdl_first_soa p12
CXF sample using WRAPPED Style in XML Binding (pure XML over HTTP)	Sample illustrates the use of Apache CXF's xml binding. This specific sample shows you how xml binding works with the doc-lit wrapped style.	wsdl_first_xml _wrapped

JAX-RS Examples

Example Name	Example Description	Example Location

JAX-RS sample collection	More then 20 JAX-RS examples	jax_rs
RESTful HTTP Binding Demo	Sample shows how to create RESTful services using CXF's HTTP binding. The server in the sample creates 3 different endpoints: a RESTful XML endpoint, a RESTful JSON endpoint, and a SOAP endpoint.	restful_http _binding

Javascript Examples

Example Name	Example Description	Example Location
Generated JavaScript using JAX-WS APIs and JSR-181	Sample shows the generation of JavaScript client code from a JAX-WS server.	js_browser _client_java _first
JavaScript Client	Sample using Document/Literal Style sample illustrates the use of the JavaScript client generator. This sample deploys the service based on the wsdl_first demo, and then provides a browser-compatible client that communicates with it.	js_browser _client_sim ple
Hello World Client sample using JavaScript	Sample demonstrates the use of (non-browser) JavaScript client to call a CXF server.	js_client
Hello World sample using JavaScript and E4X Implementations	Sample demonstrates the use of the JavaScript and E4X dynamic languages to implement JAX-WS Providers.	js_provider

WS-* Examples

Example Name	Example Description	Example Location
WS-Addressing Demo	Sample shows how WS-Addressing support in Apache CXF may be enabled.	ws_address ing
WS-Policy Demo	Sample shows how the CXF WS-Policy framework in Apache CXF uses WSDL 1.1 Policy attachments to enable the use of WS-Addressing.	ws_policy
WS-RM Demo	Sample shows how WS-ReliableMessaging support in Apache CXF may be enabled.	ws_rm
Security Interofest	Sample is being used to help implement WS-SecurityPolicy, WS-SecureConversation, and WS-Trust within CXF.	ws_security \interopfest
WS-Security (Signature and Encryption)	Sample shows how WS-Security support in Apache CXF may be enabled. WS-Security can be configured to the Client and Server endpoints by adding WSS4JInterceptors. Both Server and Client can be configured for outgoing and incoming interceptors. Various Actions like, Timestamp, UsernameToken, Signature, Encryption, etc., can be applied to the interceptors by passing appropriate configuration properties.	ws_security _sign_enc
WS-Security (UsernameToken and Timestamp)	Sample shows how WS-Security support in Apache CXF may be enabled. WS-Security can be configured to the Client and Server endpoints by adding WSS4JInterceptors. Both Server and Client can be configured for outgoing and incoming interceptors. Various Actions like, Timestamp, UsernameToken, Signature, Encryption, etc., can be applied to the interceptors by passing appropriate configuration properties.	ws_security _ut
WS-Security (UsernameToken and Timestamp)	Sample shows how WS-Security support in Apache CXF may be enabled. WS-Security can be configured to the Client and Server endpoints by adding WS-SecurityPolicies into the WSDL.	ws_security \ut_policy
WS-Security (Signature and UsernameToken)	Sample shows how WS-Security support in Apache CXF may be enabled. WS-Security can be configured to the Client and Server endpoints by adding WSS4JInterceptors. Both Server and Client can be configured for outgoing and incoming interceptors. Various Actions like, Timestamp, UsernameToken, Signature, Encryption, etc., can be applied to the interceptors by passing appropriate configuration properties.	ws_security \ut_sign

Data Bindings Examples

Example Name	Example Description	Example Location

CXF sample using code first POJO's and the Aegis Binding	Sample illustrates how to develop a service that is "code first", POJO-based. This sample uses the Aegis data binding.	aegis
CXF sample using the Aegis Binding without any webservice	Sample shows you how you can use Aegis with no web service at all (standalone) as a mapping between XML and Java.	aegis_stand alone
Hello World using Document/Literal Style and XMLBeans	Sample illustrates the use of the JAX-WS APIs and with the XMLBeans data binding to run a simple client against a standalone server using SOAP 1.1 over HTTP.	xmlbeans

CXF and JMS Examples

Example Name	Example Description	Example Location
Java First demo using JAX-WS and JMS		java_first_ja xws_jms
JMS Transport Publish/Subscribe Demo using Document-Literal Style	Sample using Document-Literal Style sample demonstrates use of the Document-Literal style binding over JMS transport using the pub/sub mechanism.	jms_pubsub
JMS Transport Queue Demo using Document-Literal Style	Sample using Document-Literal Style sample demonstrates use of the Document-Literal style binding over JMS Transport using the queue mechanism.	jms_queue

JBI Examples

Example Name	Example Description	Example Location
JBI EPEC	Sample illustrates how external CXF client using SOAP/HTTP can communicate with external CXF server using SOAP/JMS through JBI SOAP and JMS binding component (as a transformer).	integration\JBI\ext ernal_provider_ext ernal_consumer
JBI EPIC	Sample illustrates how internal CXF client that is deployed into CXF service engine can communicate with external CXF server through a generic JBI JMS binding component (as a router).	integration\JBI\ext ernal_provider_int ernal_consumer
JBI and Handlers	Sample shows how JAX-WS handlers can be used in CXF service engine. The sample consists of a CXF Service Engine and a test service assembly. The service assembly contains two service units: a service provider (server) and a service consumer (client).	integration\JBI\ha ndlers
JBI IPEC	Sample illustrates how external CXF client can communicate with internal CXF server which is deployed into CXF service engine through a generic JBI binding component (as a router).	integration\JBI\int ernal_provider_ext ernal_consumer
JBI IPIC	Sample shows how CXF can be used to implement service implementations for a Java Business Integration (JBI) container. The sample consists of a CXF Service Engine and a test service assembly. The service assembly contains two service units: a service provider (server) and a service consumer (client).	integration\JBI\int ernal_provider_int ernal_consumer

JCA Examples

Example Name	Example Description	Example Location
Inbound Connection CXF J2EE	Sample shows how to expose an Enterprise Java Bean over SOAP/HTTP using CXF.	integration\j ca\inbound
CXF Inbound Resource Adapter Message Driven Bean	Three samples... new inbound resource adapter samples (inbound-mdb, inbound-mdb-dispatch, and inbound-mdb-dispatch-wsdl). This series of inbound adapter samples leverages the JCA Specification Version 1.5 and Message Driven Bean in EJB 2.1 to activate CXF service endpoint facade inside the application server. For more information about the JCA message inflow model, please refer to chapter 12 (Message Inflow) of the JCA Specification 1.5.	integration\j ca\inbound-mdb*
Outbound Connection CXF J2EE	Sample shows how to connect with an Apache CXF Web service using a Servlet deployed in an application server; Hello World (SOAP over HTTP)	integration\j ca\outbound

		integration\jca\outbound.wsa61
CXF Outbound Resource Adapter IBM WebSphere 6.1	Sample demonstrates the new CXF outbound resource adapter.	

Miscellaneous Examples

Example Name	Example Description	Example Location
CXF Sampe using a callback object	Sample shows a client creating a callback object by passing an EndpointReferenceType to the server. The EndpointReferenceType is then used by the server to call back on the callback object.	callback
CXF Sample of Stream GZIP Interceptor	Example shows how to develop an interceptor and add the interceptor into the interceptor chain through configuration.	configuration_interceptor
Groovy with Spring Support	Sample shows how to create groovy web service implemented with Spring. You'll learn how to write a simple groovy script web service.	groovy_spring_support
Colocated Demo using Document/Literal Style		in_jvm_transport
MTOM Demo for SWA & XOP	Sample illustrates the use of a SOAP message with an attachment and XML-binary Optimized Packaging.	mtom
MTOSI Alarm Retrieval (SOAP OVER HTTP)	Sample demonstrates a simple CXF based client/server Web service implementing the MTOSI alarm retrieval service.	mtosi_1.1\alarm_retrieval
Ruby Script	Sample shows how to create ruby web service implemented with Spring. You'll learn how to write a simple ruby script web service.	ruby_spring_support
Sample usingSOAP Headers	Sample illustrates Apache CXF's support for SOAP headers.	soap_header

CORBA Examples

Example Name	Example Description	Example Location
CORBA Bank	Sample illustrates the use of JAX-WS API's for creating a service that uses the CORBA/IIOP protocol for communication. It also shows throwing exceptions across that connection.	corba\bank
CORBA Bank with WS-Addressing	Sample illustrates the use of the JAX-WS APIs to run a simple "Bank" application using CORBA/IIOP instead of SOAP /XML. It also contains standard CORBA client/server applications using pure CORBA code so you can see the JAX-WS client hit a pure CORBA server and a pure CORBA client hit the JAX-WS server.	corba\bank_ws_addressing
CORBA Hello World	Sample illustrates the use of the JAX-WS APIs to run a simple "hello world" application using CORBA/IIOP instead of SOAP/XML. It also contains standard CORBA client/server applications using pure CORBA code so you can see the JAX-WS client hit a pure CORBA server and a pure CORBA client hit the JAX-WS server.	corba\hello_world