

Jsr181ProjectPlan

Milestone: Axis .jws Deployment

Description

Annotated Java files with ".jws" file extensions are automatically registered with Axis as a Web Service and configured based on the file's metadata. Most annotations and rules specified in JSR-181 are supported.

Tasks

- Research
 - Beehive, JSR-181: done
 - Axis: done
- Checkin of initial version in to Beehive-SVN: done
- Annotation reader for class files (reflection): done
- Annotation reader for source files (apt): done
- Object Model
 - Interfaces: done
 - Applicable JSR-181 default values: done
 - Applicable JSR-181 rules: done
- Customized Axis hook, version 1: done
 - @WebService (except endpointInterface: deferred): done
 - @WebMethod: done
 - @Oneway: done
 - @WebParam: done
 - @WebResult: done
 - @HandlerChain: done
 - @SOAPBinding: done
 - @SOAPMessageHandlers: done
 - @SecurityRoles: deferred
 - @SecurityIdentity: deferred
- Backwards compatibility with non-annotated files: done

Tests

- Unit tests
 - Annotation reader for class files (reflection): done
 - Annotation reader for source files (apt): done
 - Object Model: done
 - JSR-181 default values
 - Rule: Oneway/Exceptions
 - Rule: Oneway/OUT-WebParam
 - Rule: Oneway/void
 - Rule: SOAPHandlers/HandlerChain
 - Customized Axis hook: done
 - [DropInDeploymentHandler](#): done

Documentation

- Overall architecture and design (Beehive Wiki): **work in progress** (95%)
- Sample application "AddressBook" (from Axis)
 - Server side: done
 - Simple "local" client (same VM): done
 - Simple "remote" client (WS): done
 - Documentation (incl. tutorial on how to author a .jws file with JSR-181 annotations): **work in progress** (%)

Risks

- Legal issues using JCP code
- More modifications to JSR-181 spec (moving target)
- More modifications to Axis parts interfacing with Beehive
- Inter-subproject dependencies such as shared libs, etc.
- Complex data types need more research

Milestone: [PetStore](#)

Description

Select functionality of the "Pet Store" sample application used to showcase the other two prongs of Beehive netui and controls is exposed as a [WebService](#), using JSR-181 metadata annotations. The emphasis of this milestone lies on illustrating the simplicity of authoring Web services using JSR-181 annotations compared to the more complex traditional authoring model. It would be nice to complete this milestone for [ApacheCon](#) for demonstrations.

Tasks

- Research
 - Axis: done
 - Pet Store application: done
- Update code base with JSR-181 v0.9: done
- Integrate wsm with controls
 - Use apt for compilation and verification **work in progress** (%)
 - Use reflection for deployment **work in progress** (%)
- Define demo scenario: done
 - identify [PetStore](#) functionality to be exposed through [WebService](#)
 - browser-based netui GUI used by [PetStore](#) customers for online shopping
 - stand-alone WS client w/ e.g. SWING GUI used by [PetStore](#) owner for inventory management
- Implement [WebService](#) with annotations
 - design and implement [PetStore](#) server (jws-file and supporting classes): **work in progress** (75%)
 - design and implement [PetStore](#) client (SWING GUI): **work in progress** (75%)
- Develop "Beehive story" (wsm, netui, controls): **work in progress** (%)
- Define and implement integrated sample (controls (e.g. DBControl), wsm): **work in progress** (%)

Tests

- JUnit tests for Web service - server: **open**
- JUnit tests for Web service - client: **open**

Documentation

- Describe difference between code samples using and not-using annotations: **open**
- Describe relationship between wsm and netui, controls: **open**
- Describe how to setup required deployment environment: **open**
- Optional: Documentation on how to setup development environment: **open**
- Optional: Describe setup of development environment (plugins, etc.): **open**
 - Eclipse configuration
 - Tomcat plugin
 - JDK 1.5 plugin
 - Axis source code

Risks

- Depends on availability of functional "PetStore" sample application
- Depends on other subprojects (netui, controls)
- apt still "pre-release", not (documented) programmatic interface

Milestone: JSR-181 compliant

Description

This milestone adds all the missing pieces to the implementation that are required for a full JSR-181 compliant implementation. The milestone is complete once the implementation passes the TCK.

Tasks

- Research
 - Axis: **open**
 - XMLBeans: **open**
- Customized, JSR-181-compliant Axis hook version 2
 - `@WebService.endpointInterface`: **work in progress** (50%)
 - `@SecurityIdentity`: **open**
 - `@SecurityRoles`: **open**
- WSDL to Java (generates Java server side skeletons from WSDL descriptor)
 - WSDL reader: **open**
- Java to WSDL (generates WSDL descriptor from annotated Java file)
 - WSDL writer: **open**

Tests

- Unit tests: update existing unit tests
 - WSDL reader: **open**
 - WSDL writer: **open**
- Test suite with files
 - Automatic build and deployment mechanism for test scenarios: **open**
 - WSDL verification: **open**
 - Automatic client-side code generation: **open**
 - Automatic client-side build and deployment mechanism: **open**
- Download, integrate and check in TCK: **open**
- Annotated test files that cover all functionality: **open**

Documentation

- Update samples: **open**
- Update tutorial: **open**
- Start with Java manual, samples: **open**
- Start with WSDL manual, samples: **open**
- Javadoc: **open**
- Tutorial for converting existing Web application into [WebService](#): **open**

Risks

- Depends on availability of TCK

Milestone: IDE Support

Description

This milestone provides functionality for applications that need to verify code (e.g. IDEs) through validation functionality that goes beyond validation by the Java compiler. In particular, the verification routines check for semantic errors that contradict (annotation) rules specified in JSR-181.

Tasks

- Refactor [TwoPhaseAnnotationProcessor](#) (from "controls" to "commons"): **open**
- Make JSR-181 implementation available to IDE ([TwoPhaseAnnotationProcessor](#))
 - update check(): **open**
 - update generate(): **open**
- Add appropriate error messages (text, position, etc.) for error cases: **open**

Tests

- Extended JUnit tests for [AnnotationProcessor](#): **open**

Documentation

- Use documentation from controls: –

Risks

–

Milestone: WSM Demo Application

This milestone demonstrates how the authoring model suggested by JSR-181 metadata annotations simplifies the creation of Web services. The sample application focuses on the WSM part of Beehive, which keeps the sample small and easier to understand than the more comprehensive "PetStore" sample.

Do we want to do this on top of the Pet Store sample?

Tasks

TBD.

Tests

TBD.

Documentation

TBD.

Risks

TBD.