XINS 2.0 features

101 new features of XINS 2.0.

This presentation has 102 pages

XML Schema to types

- New target *xins xsd-to-types*
- Generates .typ files with the types defined in the given XML Schema files for the specified API.

- Parameter xsd.dir is used to get the directory where the *.xsd files are located.

_WSDL Meta function

- New meta function _WSDL

- The WSDL file is included in the WAR file and returned if requested.
- WSDL is an industry standard to describe an API.
  - http://www.w3.org/TR/wsdl

XINS FF HTML Error

- Error due to XSLT will produce an HTML page containing the details of the Error instead of HTTP 500 Internal Server Error.

- The XSLT used to generate the page could be configured with the xinsff.error.page bootstrap property.

_SMD Meta function

- New meta function _SMD.
- Return the Simple Method Description of the API.
  - http://dojo.jot.com/SMD

XML to Object with mapping

- Added utility method to fill an Object with an XML Element that performs mapping of element names or attribute name.

- *Object BeanUtils.xmlToObject(Element xml, Object target, Map<String, String> elementMapping, Map<String, String> attributeMapping);*

API.getResourceAsStream()

- New method to get the InputStream of a resource located in the WAR file.
  - Improve compatibility with the different Servlet containers.
  - Avoid code duplication.
  - Useful for _WSDL, get script files.

Help in impl.xml

- When writing the impl.xml file, the order of the element matters. The order of the elements is added as comment in the generated impl.xml file when executing xins create-api.

  <!-- The order of the elements is logdoc, bootstrap-properties, runtime-properties, content, dependency, calling-convention, instance. -->

Debug fix in NetBeans menu

- Added debug fix (apply patch) in the project pop-up menu in NetBeans.
  - Debug fix allow to apply changes to the code without recreating the WAR file or restarting the server.

Tomcat deployment in NetBeans

- Added deploy-tomcat target to the nbbuild.xml
  - Useful if you want to replace the internal Servlet container with Tomcat
  - You can then use the NetBeans HTTP Monitor tools to monitor the HTTP messages received and returned when using the API.

Improved generated WSDL

- The description of the parameter and attributes has been added to the generated WSDL.

- Correct indentation of the generated WSDL.

Error parameters in WSDL

- WSDL also specifies the output parameters and output data section of the errors.

Tests with Javadoc

- The generated unit tests now also contain Javadoc.

Javadoc in Maven

- Javadoc of the XINS libraries has been added to the XINS Maven modules.
  - IDE supporting Maven will show the Javadoc with code completion

pom.xml improvements

- Improvement in the Maven generated pom.xml:
  - Added Jetty plug-in
  - Added name for the WAR file

Logging of XSLT location

- The location of the XSLT file is logged when the file is loaded and compiled in the XSLT calling convention.
  - Know which XSLT is used
  - Faster to detect the problem in case of error
  - Once cached -> no logs.

ElementParser.parse(String)

- Parse an XML String.

Conversion List/Set

- New methods to convert a org.xins.common.types.Set/List.Value to a java.util.Set/List and vice-versa.
  - add(Collection items)
  - Collection get()

Check tool more flexible

- Class fields not starting with underscores are allowed (e.g. private String description;).
- Star imports are allowed (e.g. import java.io.*;)

help-tools target

- Displays the help for the tools targets.
  - Included the required and optional parameters

smd target

- Generates the SMD (Simple Method Description) for an API.
  - xins smd
  - optional parameter: smd.endpoint

tests target

- Runs the tests of all the APIs that have tests.
  - Useful for automated testing

readFully

- Added new utility method to read an InputStream:
  - String IOReader.readFully(InputStream)
  - Remove code duplication.

_EnableAPI and _DisableAPI

- New meta functions _EnableAPI and _DisableAPI in order to disable an API for example during the maintenance of resources used by the API.

ElementParser.parse(InputStream)

- New utility method to parse XML based on an InputStream.

JMX default disabled

- JMX is by default disabled
  - org.xins.server.jmx=true to enable it
  - Faster start-up

PatternUtils.createPattern

- Utility method to compile a regular expression.
  - Catches and logs PatternExceptions
  - Avoid code duplication.

Support for default in WSDL

- If a parameter or an attribute has a default value, the default value is also specified in the generated WSDL file.

Build Java version

- The build Java version is stored in the web.xml
  - Easier to find the problem when class loading errors occur.

Optional call to the function

- You can now specify in a calling convention to not execute the function.
  - e.g. XINSFF functions defined in bootstrap properties.
  - e.g. system.describe for JSONRPC CC.
  - e.g. You want to return a specific error such as "not logged in".

test libraries

- JAR files added to the <api name>\test directory are added to the classpath when executing the tests
  - e.g. httpunit.jar

Removed alpha stuff

- Alpha DTD removed
- Old directory structure (pre 1.0) no longer supported
- _xins_old calling convention removed

Article about scripting

- New article on how to implement an API using a scripting language.
  - http://xins.sf.net/scripting.html

Confirmation for stub

- Ask confirmation before generating the stub files.
  - Prevent from removing the real implementation.

Auto-fill test forms

- Auto-fill the test form with one of the pre-defined examples
  - Faster to use when testing

Map to Object

- BeanUtils.setParameters(Map<String, ?>, Object)
  - Populates the Object with the content of the Map.
  - The key of the map is the parameter name
  - Conversion of the value is performed if needed.

Better changes detection

- Several improvements in detection of changes in the specification:
  - Shared types and error codes
  - WSDL with errors
  - OpenDoc
  - CAPI.java

Detection of XINSFF

- Detection of the XINS Front-end Framework
  - Based on command query parameter
  - or empty HTTP GET.

Indentation 4 spaces

- The indentation of the generated editable Java files is 4 spaces.
  - Java standard code convention
  - No need to configure your IDE

Clean-up logging

- Log messages have been cleaned:
  - Start-up of the API.
    - creating, created, bootstrapping, bootstrapped, initializing, initialized -> creating, bootstrapping, initializing.
    - Only the default calling convention is created and initialized.
  - When calling another API.
    - No more time-out specified unless a time-out exception occured.

Less Ant logs

- Ant target that are not executed are no longer logged.
- The fact that a Java file is not generated for a function is no longer logged.

Cleaner code in XINS

- No unneeded 'extends Objects'
- Renamed some 'String s ='
- Removed used of ThreadLocal in CallResultOutputter
- Class fields defined before the constructor
- Static methods set after the constructor

Conditional redirection

- In XINS Front-end Framework, the conditional redirection will not happen if the function returns an error.

DataElement removed

- The org.xins.client.DataElement class has been removed, replaced by org.xins.common.xml.Element
  - Can be used with BeanUtils
  - Can be stored in the XINS FF session
  - Can be edited
  - Remove code duplication

CAPI source not in Zip

- The generated source code of the CAPI is no longer included in the Zip file created by the client-<api name> target.

Download tools target

- New target: "download-tools"
  - Download third party libraries needed for the tools.
  - e.g. checkstyle.jar, j2h.jar, jmeter.jar, pmd.jar, FindBugs, ...
  - Makes it easier to run the available tools

Editable Element

- The org.xins.common.xml.Element object is now editable:
  - new Element()
  - addChild()
  - setAttribute()
  - setText()
  - setLocalName()

EnumItem in session

- EnumItem stored in the session of the XINS front-end framework are also put in the result XML.

Toolbox example

- Added new example API named toolbox.
- Implementation of the functions is done using Groovy scripting language.

Ajax DOJO toolkit example

- Added new Ajax example using the Ajax DOJO toolkit.
  - Very simple using the generated smd file.

Deprecated classes

- FastStringBuffer and FastStringWriter are no longer used as Java StringBuffer and BufferedWriter are faster.
- No longer used in XINS code, left for external APIs that might use them.

Error code public

- The generated error code classes are now public.
  - useful for BeanUtils methods.
  - useful for Spring framwork.

Unique properties

- If a runtime property is set more than once in the runtime property file the API will log it as an error and fail to start.
  - In 1.5 a warning message was logged.

Invalid value visible

- If value is invalid, the invalid value is also given in the message.
  - Easier to debug

Improved convertion

- Improved conversion in BeanUtils:
  - Date
  - Timestamp
  - Collections (List/Set)
  - Primitive types (int, boolean, ...)

Improved detection of files

- api.xml is searched only in apis/*/api.xml
- impl.xml is searched only in apis/*/impl*/impl.xml
- No longer search for environment.xml

- Before **/api.xml **/impl.xml and **/environment.xml

- Faster generation of build.xml

Better Integration with the Spring Framework

- New package org.xins.common.spring
  - XinsCapiFactoryBean
  - XinsClientInterceptor
  - XinsCapiRequestValidator
  - XinsFunctionResultValidator

JSON-RPC calling convention

- New JSON-RPC calling convention in core
  - _xins_jsonrpc
  - Support 1.0 and 1.1 specifications (automatic detection)
  - Support system.describe function

Calling conventions on demand

- Lazy creation and initialisation of the calling conventions.
  - Only the default calling convention is created and initialized when the Servlet is started
  - Faster start-up
  - Less logging (including possible exception)
  - Less memory

Limited logs

- On client side only the 160 first characters of a parameter value are logged
  - Allow to use binary types _hex and _base64 without slowing the console or filling the log file.
  - "..." is added at the end of the value.

build.properties loaded for tools

- The build.properties file is loaded when executing the tool targets
  - Allow to reuse the current settings (e.g. test.start.servlet for the code coverage)
  - Avoid to have to enter the same settings (e.g. api.name=myproject)

Object to Map

- Added utility method Map<String,?> BeanUtils.getParameters(Object)
  - Put the parameters of the object in a Map

Object to Map II

- New utility method Map<String,?> BeanUtils.getParametersAsObject(Object)
  - Gets the parameter as standard Java object
  - Useful for passing parameters to a script language that won't know XINS specific types (EnumItem, List, PropertyReader)

Object to Map III

- New utility method Map<String,String> BeanUtils.getParametersAsString(Object)
  - Put the String representation of the parameters in a Map.

SOAP CC flexible

- Methods of the SOAPCallingConvention have been changed from private to protected.
  - Depending on the framework, SOAP request for a given WSDL can be different. This feature avoid user to have to write a completely new calling convention if the SOAP messages are different than the expected ones.

Class fields before constructor

- The class fields have been moved before the constructor.
  - Fits the Java coding convention.

testforms.js

- Most of the Javascript used in the test forms has been moved to a testform.js file.
  - Avoid duplication of code
  - Smaller test form HTML pages

Static methods after constructor

- The static methods have been moved after the constructor.
  - Fits the Java coding conventions

New TextUtils methods

- Added new utility methods for text manipulation:
  - String firstCharLower(String);
  - String firstCharUpper(String);
  - String removeCharacter(String, char);

New version system

- 2.0.0 -> 2.0

Conversion

- New utility method for conversion of objects:
  - Object BeanUtils.convert(Object source, Class destinationClass);
  - Convert XINS types, Java types, String.

No exception in get methods

- Removed the ParameterNotInitializedException that was thrown in the Request.get<Param>() methods for optional parameters and attributes
  - null is returned if the parameter is not set
  - This also means that if optional the get method will return the Object type (Boolean, Integer, ...) instead of the primitive type.

Removed obsolete code

- Most of the deprecated classes and methods have been removed:
  - *org.xins.client.DataElement*, *org.xins.common.ExceptionUtils*, *org.xins.common.ant.CallTargetTasks*, *org.xins.common.text.WhislEncoding*, *org.xins.common.text.NonASCIIException*, *org.xins.common.threads.QueueTimeOutException*, *org.xins.logdoc.LogdocExceptionUtils*, *org.xins.logdoc.LogdocHexConverter*, *org.xins.logdoc.LogdocSerializable*, *org.xins.logdoc.LogdocStringBuffer*, *org.xins.logdoc.MandatoryArgumentChecker*, *org.xins.logdoc.AbstractLogdocSerializable*, *org.xins.logdoc.UnsupportedLocaleError*, *org.xins.server.Element*

Renamed XSLT property

- Renamed templates.parameter.prefix runtime property to templates.<api name>.xins-xslt.parameter.prefix.
  - Different properties for different APIs
  - More consistent with XINS Front-end Framework

Renamed XSLT property II

- Replaced the templates.callingconvention.source property of the _xins-xslt CC to templates.<api name>.xins-xslt.source.
  - Different properties for different APIs
  - More consistent with XINS Front-end Framework

Reorganized WAR file

- The generated WAR file has been reorganized:
  - Removed impl.xml file.
  - Added WSDL file.
  - specs moved to the WEB-INF directory.

Removed unneeded comments

- Removed comments that separated the different sections:
  - In generated code and in XINS sources

Logging exception simplified

- Removed unneeded parameters to the call to Utils.logIgnoredException and Utils.logProgrammingException.
  - Automatic detection of the class/method throwing the exception and of the class/method logging the exception.
  - Cleaner code

REST example

- Added pure REST example
  - REST is Representational State Transfer
  - http://en.wikipedia.org/wiki/Representational_State_Transfer
  - RESTCallingConvention

Several bug fixes

- Several bugs have been fixed in this release.
  - The most important ones have been integrated in XINS 1.5.2.

Servlet container improvements

- Several improvements done for the Servlet container.
  - better path support for cookies and Servlets

Shared error code

- Added shared error code.
  - Works as shared types

No stack trace for ignorable exception

- The stack trace is no longer logged when logging an ignorable exception.
  - smaller logs
  - more readable, easier to detect real problems

Coverage without server

- Starting the Servlet container for the test coverage is now optional.
  - Execute the code coverage with the API running in Tomcat, Jetty or your J2EE server.

Keep environment in test forms

- Test forms remember the last environment chosen.
  - When moving from one test form to another one, the last chosen environment is stored in the cookies.

Rebuild war when testing

- test-<api name> target depends on war-<api name> target if the server should be started.

Dependencies in test classpath

- The API dependencies are added to the classpath for compiling the tests.

Full path for build directory

- The full path is used to specify the build directory used to generate and compile and the files.
  - Could be specify to somewhere else. -Dbuilddir=...
  - Fixed some problems when the script is executed from another directory like within IDEs.

Calling convention extendable

- The calling conventions are now public and extendable.
  - Easier to create a calling convention similar to an already existing one.

HTML for Control command

- The Control command of the XINS front-end framework will return an HTML page
  - Also if an Control action is executed

Location of created logdoc

- The create-logdoc target now informs where are the files created.

Public check method

- The FunctionResult.checkOutputParameters() method is now public in order to enable external validation.

Indented WSDL

- The generated WSDL file is correctly indented with tabs.

public get method

- The get methods of the FunctionRequest object are now public.
  - Usable with BeanUtils or Spring Framework for example.

Unix xins as xins.bat

- The Unix xins script now uses make-build.xml to create the build.xml file instead of bootstrap.xml.

Version printed if known

- The version of the API is printed in the footer in the specdocs only when known.
  - Avoid to have “version ?.?”

Updated libraries

- Updated libraries:
  - commons-logging (1.1)
  - Windows installer (better Vista support)
  - Sarissa (Ajax lib)

mode=source forwarded

- When mode=source is used in XINS FF, the redirection also forward this parameter.
  - Easier for (unit) testing
  - Easier for debugging

WSDL to API

- Generates the specification of an API based on a WSDL.
  - Tested with a lot of different WSDL (.Net, Axis, ...)
  - The input can be a WSDL file or a WSDL URL

Yahoo! JSON calling convention

- Added new JSON calling convention
  - Yahoo! JSON calling convention
  - Includes support for callback
  - Automatic detection if *output=json*
  - Easy to use with GWT (Google Web Toolkit)

Fixed _ReloadProperties

- The _ReloadProperties meta function did not work when the watching interval was higher than 0. This is fixed.

Performance

- Internal Servlet container 50% faster
- Parameters serialized only if logged
- No parsing of the specs to know if a error code is functional or technical
- Optimized HTTPServiceCaller