

WS Core Specification Upgrade

1. Introduction

The Java WS Core and C WS Core components of the Globus Toolkit are implementations of the Web Services Resource Framework (WSRF) and the Web Service Notification (WSN) family of standards. They provide APIs and tools for building stateful Web services, with SOAP over HTTP or HTTP with TLS as transport protocol. Java WS Core builds on Apache Axis 1.2, a SOAP 1.1 engine.

These components implement OASIS WSRF/WSN working draft specifications from June 2004, with minor fixes to the 1.2-draft-01 published schemas. These specifications are based on the March 2004 version of the WS-Addressing specification.

This document presents an analysis of approaches to support the final WSRF/WSN specifications in Java WS Core and C WS Core.

2. Support for WSRF 1.2 and WSN 1.3

In April 2006 WSRF version 1.2 and WSN version 1.3 were accepted as an OASIS Standard. Some changes have been made to the specification from the draft version that is supported by GT 4.0. Details on the differences are provided in Section 4. In summary, the specification changes cause no loss of functionality, but alter wire message formats and provide optional additional functionality.

The Java WS Core and C WS Core components can be enhanced to support final versions of the WSRF and WSN specifications. This change would allow GT to host and interact with WS-compliant services. In particular, this allows GT to support OGF's OGSA WSRF Basic Profile and OGSA WSRF Basic Security Profile. The upgrade will also aid interoperability with other frameworks that use these standards. A list of OGF specifications and profiles that use the new standards is provided in Section 6 of the document.

The toolkit can add support for these new specifications either as an in-place replacement of the existing WSRF implementation or as a second implementation that core supports.

2.1 *Support final specifications only*

If an in-place replacement is done, the current operation providers and core APIs will be updated to use the latest specification. The SOAP handlers that parse the addressing headers will be modified to support WS Addressing 1.0. Only minor API changes are expected for this implementation.

Web service interfaces are defined by WSDL documents. These documents include XML schema documents which define data types and WSDL service definitions. Each web service's interface will require some change to conform to the WSRF and WS-Addressing standards. Service WSDLs will need to be modified to import the new WSRF and WS-Addressing

schemas. ResourceProperty definitions, WSRF operations, and Addressing-related elements and attributes will need to be updated in each service WSDL. It is possible that the WSDL updates could be done automatically.

From a user's perspective, this option would break backwards compatibility with old clients and servers, that is, new clients cannot interoperate with old services and vice versa. Clients would need different EndpointReference documents to know how to contact services using the new standards, and the SOAP messages sent to and from the services would be different as well.

From a service developer's perspective this would require updating of all service WSDLs to use the standards as described above. Java service deployment descriptors would need to be changed to use new operation providers for additional features in the latest specification. The Java WS Core API is not expected to change, but the functionality will be replaced with new specification implementation and hence the port effort is expected to be minimal. Minor C WS Core API changes are expected for code which creates and modifies EndpointReference data and manages WSNT subscriptions.

2.2 Support both sets of specifications

If the final version of the specifications is added without removing support for current specifications, then the containers would be able to host and contact services that use draft specifications as well as services that implement the final specifications. The model being proposed is to treat each set of specification as an operational universe, that is, each service endpoint would only be capable of processing requests which use a particular set of specifications. A service implementer who wishes to interoperate with multiple operation universes would need to develop multiple WSDLs and service deployment descriptors. These could share internal application logic. Interfaces for these services, including service definitions, fault messages, factory operations may be different for the various universes.

In this scenario, a new set of operation providers and internal APIs will be provided for the final specifications, while the old ones will remain as is without any public API change. A SOAP handler to parse the new addressing specification will be added and, based on the version of addressing used in the SOAP messages, the framework will detect the universe to operate in. Bindings for different versions of data types used by the different WSRF and WS-Addressing specifications will be included in the toolkit, in addition to multiple APIs for managing different EndpointReference types and notification message types.

From a user's perspective, existing client/service interaction will not change.

From a service developer's perspective, the developer must choose the correct WSDL elements and attributes and operation providers for the service based on the specifications used. Also developers must use the correct API for processing specification dependent objects like EndpointReferences and Faults.

3. Questions

- ◆ Do you need final version of WSRF/WSN specification for your project?

- Which OGF Profiles do you care about?
- What are the OGF specifications your project uses?
- ◆ Do you interoperate with other WSRF/WSN implementations? What version of specifications do they use
- ◆ Do you need to host services with draft version of specification with services that use final specification?
 - Can you host two separate containers for draft and final version of services?

4. Specification Changes

4.1 WS Addressing specification

WS-Addressing provides mechanisms to describe Web services and messages. GT contains an implementation based on the March 2004 draft of WS-Addressing. Since that time, a new version has been submitted and consequently approved as a W3C recommendation (W3C WS-Addressing 1.0). Since its approval, WS-Addressing 1.0 has been adopted as a part of many other web service standards. Related specifications that use WS-Addressing 1.0 are listed in the next section.

The following is a brief summary of differences between the two WS-Addressing versions.

Table 1 WS-Addressing Differences

Version	March 2004	W3C WS-Addressing 1.0
Namespace	http://schemas.xmlsoap.org/ws/2004/03/addressing	http://www.w3.org/2005/08/addressing
References	ReferenceProperties child of EPR	ReferenceParameters child of EPR
EPR Metadata	PortType and ServiceName children of EPR	Children of MetaData child of EPR (extensible)

4.2 WSRF 1.2

The final version (1.2) of the specifications was published as OASIS standard in April 2006 and uses WS Addressing 1.0 described in the previous section. A list of OGF specifications/profiles that use this version of the specification are listed in the next section. For all of these specifications, all WSDL messages are decorated with Action attributes from the W3C WS-Addressing 1.0 namespace. The following is a summary of differences from the draft version implemented in GT and the 1.2 standard versions.

4.2.1 WS-Base Faults

The WS-BaseFaults specification defines a base set of information which is present in faults generated by WSRF-compliant services. These faults include the address of the fault originator, the time when the fault occurred, and, optionally, a chain of causal faults. The fault structures

used in this specification are extended and used in all of the other WSRF standards documents. The full specification for WS-BaseFaults is available at http://docs.oasis-open.org/wsrf/wsrf-ws_base_faults-1.2-spec-os.pdf.

Table 2 WS BaseFaults Differences

Version	WS-BaseFaults 1.2 draft 2	WS-BaseFaults 1.2
Namespace	http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-BaseFaults-1.2-draft-01.xsd	http://docs.oasis-open.org/wsrf/bf-2
Fault Extensibility	Any content after ErrorCode with mandatory dialect attribute	Any elements at head of fault type
Fault Originator	March 2004 WS-Addressing EPR	W3C WS-Addressing EPR

4.2.2 WS-Resource Lifetime

The WS-ResourceLifetime specification defines operations and resource properties which can be used to manage soft-state lifetime of a WSRF resource. The resource properties defined by this standard are defined in terms of WS-ResourceProperties defined below.

Table 3 WS-ResourceLifetime 1.2 Differences

Version	WS-ResourceLifetime 1.2 draft 3	WS-ResourceLifetime 1.2
Namespace	http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceLifetime-1.2-draft-01.xsd	http://docs.oasis-open.org/wsrf/rl-2
TerminationTime schema type	dateTime	dateTime or duration

4.2.3 WS-ResourceProperties

The WS-ResourceProperties specification defines operations and WSDL attributes for defining how to expose properties of a WSRF resource. In addition to minor changes to the schemas since the draft used in GT 4.0, several new features are added to the specification.

Table 4 - WS-ResourceProperties differences

Version	WS-ResourceProperties 1.2 draft 4	WS-ResourceProperties 1.2
Namespace	http://docs.oasis-open.org/wsrf/2004/06/wsrf-WS-ResourceProperties-1.2-draft-01.xsd	http://docs.oasis-open.org/wsrf/rp-2
Properties Document Type	ComplexType containing a sequence or all	ComplexType

4.2.3.1 GetResourcePropertyDocument

This WSDL portType defines an operation for retrieving the entire Resource Property document associated with a WSRF resource. This can be approximated in draft 4 by invoking the QueryResourceProperties operation with the query expression “//*”.

4.2.3.2 PutResourcePropertyDocument

This WSDL portType defines an operation for replacing the entire Resource Property document associated with a WSRF resource. This can be approximated in draft 4 by invoking SetResourceProperties with a set of Delete and Update child elements.

4.2.3.3 InsertResourceProperties

This WSDL portType defines an operation for inserting resource property values into a WSRF resource’s properties document. This can be approximated in draft 4 by invoking SetResourceProperties with Insert sub elements. However, services are unable to declare in a WSDL that only Insert (vs Update or Delete) is supported by the service implementation.

4.2.3.4 UpdateResourceProperties

This WSDL portType defines an operation for replacing resource property values in a WSRF resource’s properties document. This can be approximated in draft 4 by invoking SetResourceProperties with Update sub elements. However, services are unable to declare in a WSDL that only Update (vs Insert or Delete) is supported by the service implementation.

4.2.3.5 DeleteResourceProperties

This WSDL portType defines an operation for removing resource property values from a WSRF resource’s properties document. This can be approximated in draft 4 by invoking SetResourceProperties with Delete sub elements. However, services are unable to declare in a WSDL that only Delete (vs Insert or Update) is supported by the service implementation.

4.2.3.6 AnyResourcePropertyValueChange

This topic, if provided a service, allows a subscriber to be notified if any resource property value change occurs (Insert, Update, Delete)

4.2.4 WS-ServiceGroup

The WS-ServiceGroup specification defines operations and resource properties for defining how collections of WSRF resources and services. The draft used in GT 4.0 is very close to the 1.2 final specification, with the differences being mostly namespace changes (for the service group namespaces and it's dependencies on other WSRF specifications).

Table 5 - WS-ServiceGroup Differences

Version	WS-ResourceProperties 1.2 draft 2	WS-ResourceProperties 1.2
Namespace	http://docs.oasis-open.org/wsr/2004/06/wsr-WS-ServiceGroup-1.2-draft-01.xsd	http://docs.oasis-open.org/wsr/sg-2

WSDL documents which define WSRF-compliant services typically use cut-and-paste techniques or automated WSDL preprocessing to include WSRF-related operations and data types in their service definitions. These WSDL documents would need to be updated to include the appropriate definitions.

4.3 WS-Base Notification and WS-Topics

The WS-BaseNotification specification underwent the most changes since GT 4.0 was released. Most of the changes are to the Topic Set model, a re-factoring of the subscription message, and the addition of the PullPoint port type. Like the WSRF standards, the WS-BaseNotification standard includes references to W3C WS-Addressing 1.0. Additionally, it defines WSRF Resource Properties related to topics that a notification producing service can associate with a resource.

Table 6 - WS-BaseNotification Differences

Version	WS-BaseNotification 1.2 draft 3	WS-BaseNotification 1.3
Namespace	http://docs.oasis-open.org/wsn/2004/06/wsn-WS-BaseNotification-1.2-draft-01.xsd	http://docs.oasis-open.org/wsn/b-2
Subscribe to Resource Properties	Subscribe to Topics whose name matches the Resource Property	Subscribe using ProducerProperties filter

Subscription Message	Includes UseNotify, Precondition, Selector, SubscriptionPolicy	Includes Filter
-----------------------------	--	-----------------

4.4 PullPoint

The PullPoint interface is a new feature in WS-Notification 1.3. It provides a way for clients which are unable to receive notifications to request a third party queue notification messages and allow them to be requested by the client. This solves (in some sense) the firewall problem for clients which would otherwise be unable to receive notifications.

4.5 WS-Topics

The Topic Set model defined in the WS-Topics specification changed to include a different nomenclature for naming Topic and a slightly different model for referring to Topic Trees which include topics in multiple namespaces. The GT services use Simple Topics, which do not include any hierarchical Topic definitions, so this change does not affect the toolkit.

The Subscription message format has been updated to collect various subscription-topic related information into a Filter element. Filters allow clients to subscribe to multiple Topics at once and to subscribe to Resource Properties without requiring a Topic Set definition for each property.

5. Related Specifications

This section lists some higher-level Specifications and profiles that are relevant to the development of the Globus Toolkit, especially in the field of interoperability with other web service implementations.

5.1 WS-I Basic Profile 1.0

The WS-I Basic Profile describes a set of standards and messaging conventions to support interoperable web services implementations. The GT 4.0 implements some of these recommendations, but fails in its support of WS Addressing. WS-I Basic Profile 1.0 has wide support in the web services community.

5.2 OGSA WSRF Basic Profile 1.0

The OGSA WSRF Basic Profile 1.0 describes a standard profile for developing Grid Services. This standard is produced by the OGF. Other OGF specifications reference this standard as a way to provide simple interoperable definitions of grid services. Many OGF recommendations build on this profile as a standard rendering of how the service bindings should be rendered.

5.3 OGSA Basic Security Profile 1.0 - Core

The OGSA Basic Security Profile 1.0 describes a standard profile for web services to publish the public key associated with a service. This profile is based on the W3C WS-Addressing standard. Public keys are included in service EPRs.

5.4 *WS-Transfer*

WS-Transfer is an emerging standard for defining stateful web resources, similar to WSRF. However, it is based on a more REST-like model, which includes a much more limited set of the operations on resources (Create, Get, Put, Delete). Like WSRF, this standard is based on W3C WS-Addressing 1.0.

5.5 *Other OGF Service Standards*

Various OGF working groups are producing (or have produced) standards which include service definitions which include references to OGSA WSRF Basic Profile. Some communities have expressed interest in having GT services which provide similar services be made compatible with these specifications. Some specifications include:

- OGSA Basic Execution Service 1.0 (draft 33)
- Byte IO (Data Transfer)
- Web Services Data Access and Integration – The Core (WS-DAI) Specification, Version 1.0
- WS-Naming (draft 6)
- Web Services Agreement Specification (WS-Agreement)