IBM

Migrating from WSRF to WSRT

Level: Intermediate

David A. Cafaro (dac@cafaro.net), Computational Researcher, Consultant

31 Jul 2007

The Web services group has migrated from the original WS-Resource Framework (WSRF) standard to the WS-ResourceTransfer (WS-RT) framework. WS-RT combines elements of the original WSRF standard with the WS-Management standards to enable easier exchange of resource information and objects between different components. We'll examine the two standards, the differences between them, and how to move and migrate between the two — for compatibility and to help you migrate to the new standard for interoperability with other applications.

Introduction

With the growth of networking and the idea of distributed computing load, the concept and implementation of grid computing has matured. Though the definition of "grid computing" can mean different things to different groups, it generally refers to the interoperability and communication of heterogeneous resources over a network in a structured or managed system. One way to implement grid infrastructure is through the use of Web-based services. WS-Resource Framework (WSRF) is one specification toward the implementation of Web-based grid services. Coupled with the WS-Notification (WSN) specifications, these provided an excellent starting point for Web services-based grid computing.

At around the same time, competing standards, such as WS-Transfer, were also being developed and deployed. Since these were competing standards and did not communicate with each other, the idea of networking separately developed grid resources was being hampered.

In March 2005, the developers of WS-Transfer and WSRF announced a plan for a new specification: WS-ResourceTransfer (WS-RT). WS-RT was an expansion and enhancement of WS-Transfer that added the functionality WSRF provided that the original WS-Transfer did not. Specifically, WS-RT added functionality from WS-ResourceProperties and Web Services ResourceLifetime (WS-ResourceLifetime) — two components of WSRF that had certain detailed control mechanisms that were lacking in WS-Transfer.

In this article, we'll look at what functionality is provided by WSRF — specifically, WS-ResourceProperties and WS-ResourceLifetime — what functionality is provided by WS-RT, how they overlap, and what migration strategies you might take to move WSRF to WS-RT.

WS-Resource Framework (WSRF)

The WSRF specification consists of four primary components: WS-ResourceProperties, WS-ResourceLifetime, WS-ServiceGroup (WSRF-SG), and WS-BaseFaults (WSRF-BF). Of these component specifications, WS-ResourceProperties and WS-ResourceLifetime are the primary concerns when looking at a migrations to WS-ResourceLifetime. These two components deal primarily with resource object representation and access. They provide the key specifications for Web services to interact

with each other's resources. Below is a brief description of what each component of the WSRF specification standard provides.

But first, let's look at WS-ResourceProperties, one of the two primary focuses of this article.

WS-ResourceProperties

WS-ResourceProperties deals with the properties associated with a given resource and the characteristics of that resource. The WS-ResourceProperties specifications document standardizes how services may present the definition of a resource's properties within a Web services architecture. By using these specifications, you can present a standardized view of what resources are available and how they may be accessed through the WSRF. WS-ResourceProperties provides a set of standard messages a requester can use to query and update the property values within a published resource. These messages are restrained to subsets based on what resource properties are defined for a given service and resource. See Resources for further information.

WS-RT used some of the expanded abilities of `get`, `put`, `create`, and `destroy` statements within WS-ResourceProperties to provide for fragmented access to resource documents. This fragmented access will be discussed in detail. Next, we will look at WS-ResourceLifetime, the second of the two WSRF specifications related to WS-RT.

WS-ResourceLifetime

WS-ResourceLifetime deals with how a give resource may be destroyed (deleted) and what constraints are placed on this "lifetime" of a resource. The lifetime restrictions can include the ability to immediately destroy a resource or schedule time-based destruction. The lifetime restrictions are placed within WS-ResourceProperties so they can be queried. See Resources for further information.

WS-RT added the concepts of lifetime restrictions to resource properties based on the specifications of WS-ResourceLifetime. This provides greater control over temporal aspects of resource listings than previously provided by WS-Transfer alone.

WS-SG and WS-BF are touched on here, but are not as critical to our discussion of WSRF to WS-RT migrations. Both specifications are inherently covered by the core specifications and are redundant to some degree.

WS-Service Group (WSRF-SG)

Though not a focus of this article, WS-SG is part of the WSRF also. WSRF-SG focuses on creating heterogeneous collections of Web services and resources. Service groups are represented by components called *entries*, which are themselves considered resources. In a way, a service group is a subgrouping of multiple resources based on WSRF presented within a top-level WSRF framework. Service groups can contain most of the same properties and objects that would be available to the individual WSRF-compliant resources that have been collected within the group. See Resources for further information.

WS-Base Faults (WSRF-BF)

WSRF-BF is another specification that isn't a focus of this article. WS-RT provides fault standardization, which covers the original WS-Transfer specification, as well as the enhancements made to WS-RT. WSRF-BF provides a simplified standard set of fault information that may be reported to help resolve problems within the Web service. WSRF-BF also provides rules for the extension of these standard faults, as well as basic use by a Web service. See Resources for further information.

Now that we have covered what areas of WSRF we will be focusing on, as well as what they provide, we can look at what the new WS-RT specification covers and how that relates to the old WSRF specifications.

WS-ResourceTransfer (WS-RT)

WS-RT was born out of (and, hence, an enhancement of) WS-Transfer. The developers of WS-RT looked at WS-Transfer and WSRF, and added the functionality found in WSRF missing in WS-Transfer to WS-RT. Through this, they were also able to provide the basics needed to ease the migration from either of these older standards to the new WS-RT. Since WS-RT was just an enhancement of WS-Transfer, WS-Transfer functionality is preserved as it was originally. Though exact operational matches for WSRF may not exist, the functionality to translate those operations does exist within WS-RT.

The WS-RT specification covers the definition of resources, as well as how to access that information. It provides the details of how to present storage resource, computational resources, or other resources and time constraints associated with those resources. Beyond the original `get`, `put`, `create`, and `delete` operations on resources provided in WS-Transfer, WS-RT expands on those and adds the functionality of "fragment access." This allows an operation to access a small part of the overall XML resource representation. This was functionality provided for in WSRF that wasn't available in WS-Transfer. There has also been a reworking of the metadata support to provide for a lifetime element that contains information similar to what is defined in the WS-ResourceLifetime specification.

For further details on what WS-RT provides and how it is implemented, read the "Meet the Specs" series (see Resources). These articles provide great detail into what has been done in the WS-RT specification and how you would implement them.

With this basic foundation of the WSRF and WS-RT specifications, we will now look at two possible migration techniques that can be used to allow for the migration to the new WS-RT framework.

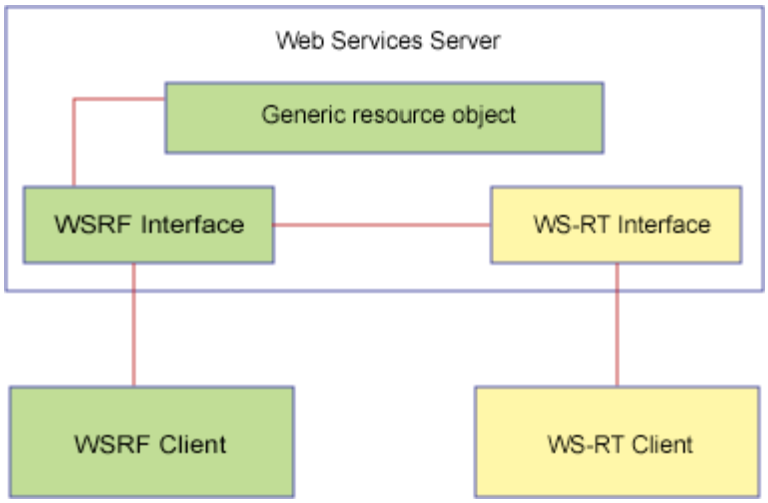
Migrating WSRF to WS-RT

There are two primary methods to provide for migration from WSRF to WS-RT. First is by providing access to both WSRF and WS-RT via delegation. The second is the use of XSLT to directly translate WSRF to and from WS-RT. In the following sections, I will provide some details on how these might be implemented. See Resources for further information.

Dual services via delegation

In this arrangement, you can run WSRF- and WS-RT-based services at the same time since WS-RT operations can be almost completely mapped to equivalent WSRF operations. In Figure 1, you will see a simple diagram of how this is set up.

Figure 1. Example of deleted migration setup



You can see how WSRF clients can continue to use your existing WSRF services. WS-RT clients will use the WS-RT services provided by the server, which are mapped within the server to the WSRF operations service. This allows the server to service WS-RT and WSRF clients during your transition period. For future development, it is recommended to allow your core resources to be defined in a general manner your application stack can provide to your WSRF. This way, as you later prepare to phase out your old WSRF clients, you can eventually allow your WS-RT services to directly interact with your resources with out retooling your core resources. For a good example of this method, check out Apache Muse (see Resources). Under the preview tree of the project, you will find a reference implementation of WS-RT delegation.

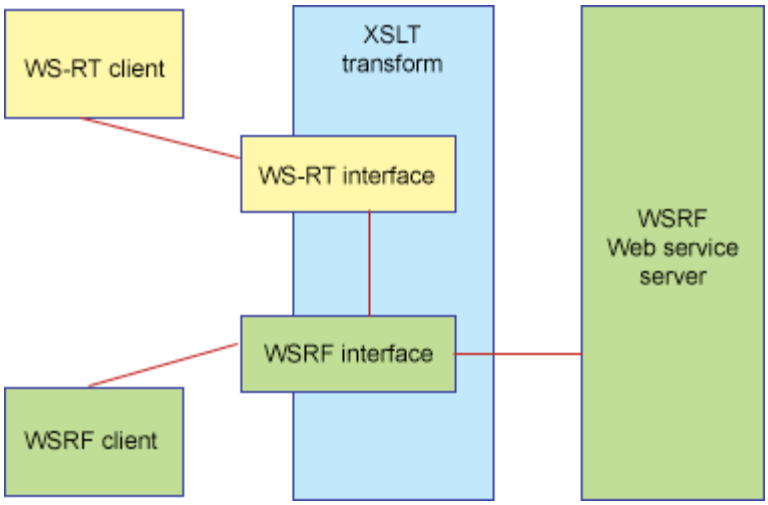
During this transition period, you would have some redundancy in your published properties and operations. You would have both WSRF and WS-RT published, and nonoverlapping specs, such as `wsrl:Destroy()` (from WSRF) and `ws-t:Delete()` (from WS-RT), which would be providing similar functionality, but to different clients. As your clients migrate to only WS-RT, you would eventually eliminate the publishing of WSRF-specific items.

The disadvantage of this is that you must code a delegation layer into your service engine if your service does not provide for the delegation model. If your engine supports a delegation model, you need only add the module to your existing service.

Translated service via XSLT

Another method for migration is the use of XSLT. Though an XSLT transforms WS-Transfer, WSRF and WS-RT clients can be serviced by a Web service regardless of whether it is a WSRF or WS-RT server. Figure 2 shows how this might be set up.

Figure 2. Example of XSLT transform setup



As shown above, using XSLT to transform the service requests, you can maintain your current WSRF Web service while allowing the migration to new WS-RT-based clients. WSRF clients can transverse the XSLT layer with no translation. WS-RT clients transverse the XSLT layer via an XML-to-XML translation to WSRF. The reverse can also be done, providing translation from WSRF clients to WS-RT Web services.

There are two primary disadvantages to this approach. First, you must code a translation layer and service to provide the XSLT transform between your clients and your Web service. Second, there are limits to when this can be applied. The XML-to-XML transform requires that there be a direct 1:1 mapping in both specifications (WS-RT and WSRF) for the given transforms.

For more details on XSLT transforms vs. the delegation system, take a look at "WSDM/WS-Man Reconciliation" (see Resources). This document provides a detailed look at the issues involved that are beyond the scope of this article.

Summary

We have provided a brief description of the WS-ResourceTransfer (WS-RT) and related WS-Resource Framework (WSRF) specifications in preparation for looking at the issues involved in planning a migration strategy. We then looked at two possible strategies for dealing with migration from the WSRF to WS-RT. These two strategies, XSLT transforms and delegation, provide a method to deal with most possible migration situations. Though neither will solve every situation, by studying your individual needs and future plans, one of these — or a combination of them — should set you on the path to a smooth transition to WS-RT-based Web services.

Resources

Learn

- OASIS Web Services Resource Framework Technical Committee Documentation provides current OASIS WSRF V1.2 specifications and service descriptions. The entire WSRF standard, as well as a WSRF Primer and several other helpful links, are available.

- "WSDM/WS-Man Reconciliation" gives a detailed overview of what's new in WS-RT vs. WS-Transfer vs. WSRF and covers migration strategies in detail.
- See what Ian Foster has to say about the release of the WS-RT specification in his WS-ResourceTransfer Specification Released blog entry.
- Ian Robinson, the editor of the published WS-RT specification, gives an update on the WS-RT specification in his WS-ResourceTransfer Update blog entry.
- For an introduction to the WS-ResourceTransfer specification, read the "Meet the specs" series on developerWorks: Intro to WS-ResourceTransfer 1.0, WS-RT 1.0 operations, Part 1, Part 2, and Part 3.
- Be sure to check out the "Building a grid using Web services standards" series on building a grid more focused on WSRF.
- The developerWorks SOA and Web services zone provides guides and tutorials on Web services and Web services standards.
- To listen to interesting interviews and discussions for software developers, check out check out developerWorks podcasts.
- Stay current with developerWorks' Technical events and webcasts.
- Check out upcoming conferences, trade shows, webcasts, and other Events around the world that are of interest to IBM open source developers.
- Visit the developerWorks Grid computing zone for more grid information.

Get products and technologies

- Visit WS-Resource Transfer to download the latest WS-RT specification, as well as the WS-RT WSDL and WS-RT XSD documents.
- Check out the full Web Services Transfer (WS-Transfer) and the Web Services ResourceTransfer (WS-RT) specifications.
- See the WSDL and schema for Web Services Resource Transfer (WS-ResourceTransfer).
- Visit the Globus Alliance, a community of organizations and individuals developing fundamental technologies behind the grid.
- Check out the Web Services Project at Apache for a collection of Web services projects, including WSRF and others.
- Apache Muse provides one of the Apache Web services projects. An initial implementation of WS-RT is also provided under the preview tree of the 2.2.x version of Muse.
- Don't miss the IBM alphaWorks Interoperability Demo for Converged Web Service Management Specifications.
- Download IBM product evaluation versions, and get your hands on application development tools and middleware products from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.

- Innovate your next development project with IBM trial software, available for download or on DVD.

Discuss

- Participate in developerWorks blogs and get involved in the developerWorks community.

About the author

David A. Cafaro is currently employed at Georgetown University's Advanced Research Computing (ARC) group, where he supports computational research needs through the use of Linux and open source technologies and is a Red Hat Certified Engineer. In addition, he sits on the Program Committee for LinuxWorld Expo and Summit ,where he helps develop track content and technology focus. He has worked as a security analyst at Tresys Technology with a focus on SELinux policy work. He is active in the open source and Linux communities as a member of the Tux.org Board of Directors.