

# Web Services Test Forum (WSTF): Bridging the gap between promises and reality

Skill Level: Intermediate

[Doug Davis \(dug@us.ibm.com\)](mailto:dug@us.ibm.com)  
Software Architect  
IBM

08 Dec 2008

SOAP-based Web services have come a long way since their creation many years ago. Recently, the number of new specifications being developed has slowed quite a bit, and this is allowing the community time to settle down and take a closer look at the base infrastructure that has been developed. Have the promises of Web service interoperability been met? Do the Web service specifications really work *out of the box* as they should? This article addresses these questions and introduces the Web Services Test Forum (WSTF). WSTF is a new community-based forum aimed at addressing interoperability issues with Web services.

## State of the WS-\* union

While Web service specifications sometimes do work as expected, if you use them on a regular basis, you have probably experienced far too many times when they don't. It would be easy to spout the usual complaints about what's wrong: the specifications are too complex, there are too many of them, there are too many options, the specification writers and coders were resource constrained, and so on. While each of these might have some validity, in many cases the same general comments may also apply to many of the other projects you work on. This article will try to go a bit deeper and look at some of the underlying issues related to Web service interoperability.

Most WS-\* specifications have followed the same process:

1. Develop a new WS-\* specification.

2. Do some interoperability testing.
3. Take the specification to a standards organization.
4. Do some more interoperability testing.
5. Stamp it as done.

This process has served relatively well, but something must be missing when even the core set of specifications do not always *work out of the box* for our customers.

This article explores what some of those missing pieces might be and introduces a new initiative called the Web Services Test Forum (WSTF). The goal of the WSTF is to bring together participants from across the Web services community and provide a forum for them to discuss and tackle the shared challenges they are facing related to the interoperability of specifications.

## Barriers to success

Three things that could contribute to the ease of use and interoperability of Web services, but that are missing from the current process, are:

- Customer involvement
- A long-term testing strategy
- A formal community to share ideas and concerns

### Customers fending for themselves

One of the key things missing from the current process is the customer. While specification writers may claim to be keeping customers and their scenarios in mind, something is clearly missing when a fair number of customers can't get even the most basic stuff to work without some amount of frustration.

There are several aspects of meeting the customers needs to consider. The most basic is simply meeting customers' functional needs (for example, ensuring SOAP gets message X from point A to point B reliably). Beyond that, there are smaller and less tangible aspects to meeting customer needs.

Examples of these other needs include things like customers having a place to ask more than one Web service vendor how things are supposed to work—mainly so that the questioner can get an unbiased answer. Or, being able to quickly see what several implementations do with common usage patterns, without having to call each vendor on the phone and ask them to devote days (or weeks) to set up a

testing environment.

These smaller and less tangible customer needs actually combine to play a large role in ensuring that the Web services community is meeting its objectives. Unfortunately, the community often overlooks these needs.

### **Test strategies missing the big picture**

Another possible barrier to success is how the interoperability and verification process itself has been set up. Is the purpose of the testing to validate that what's in the specification can be sent and consumed by SOAP endpoints, or is it to validate that the specification composes with the rest of the Web service stack and actually meets the needs of customers? In the past there has been too much focus on the former and this is why there are usually very few interoperability issues found during the testing events.

When the focus is simply on testing the XML that flows on the wire and not on the higher level business problem trying to be solved, it's too easy to miss the bigger picture. As a result, when customers go beyond the tightly scripted and tested scenarios, things tend to break down rather quickly—even if the usage they have in mind is a trivial one.

Additionally, there are inherent problems with the *point in time* nature of the current interoperability testing. In the process outlined above, the specifications go through a number of interoperability testing efforts. These efforts may be fine for the moment, but what measures are in place to ensure that after those events are completed things will continue to work? Reproducing these interoperability environments can be time consuming and often requires knowledge of other vendors' products. These two factors limit the success and cost effectiveness of this approach as a long term solution.

### **No place for unbiased discussion**

Please note that the above is not meant to be a criticism of what has gone on in the past. The hard work of many individuals and groups has resulted in many good specifications. The very fact that so many people from so many different companies and organizations have come together to work on the development of the standards is itself a huge success. However, like most things, a little shake up in the status quo could improve things.

Something that the Web services community has been lacking is a place where all members in all various roles can come together to discuss testing and other concerns related to Web services. For example, there has been no place that a Web service vendor, an ISV, or a customer, could go to ask a broad community of people a question. Most other places have a natural bias. An inquiry posed to a particular vendor would naturally tend to result in an answer skewed towards that vendor's

own product's capabilities.

## A new initiative

To help address some of the current issues with Web service interoperability testing and break down the barriers described above, a new initiative has been started: the Web Services Test Forum (WSTF). WSTF is a community-based forum that follows a model similar to that of open source communities. Participants are free to suggest new interoperability testing scenarios as well as test scenarios that they view as important. This is a simple enough idea, but the following details of the group should help explain how this improves on the way things have been done in the past.

The WSTF was originally formed to perform scenario-based interoperability testing that was not being done elsewhere. In particular, this was to be compositional testing where a group of specifications were tested together to ensure that they really were as composable as advertised. However, before these more advanced combinatorial tests could be run, it had to be verified that: 1) the individual specifications themselves worked, and 2) the proposed scenario wasn't just a single developer's dream, and that it was based on actual customer needs.

From this simple idea, the WSTF was born. At its most basic level, the WSTF is a place for all members of the Web services community to discuss testing and any other concerns related to Web services.

## A community with a focus on testing

Simply providing a forum for people to ask questions and discuss ideas isn't going to help solve interoperability issues. The forum needs to have a focus and purpose. That is the value provided by the WSTF's customer-focused scenarios.

Unlike previous interoperability efforts, the WSTF is specifically designed to verify and examine scenarios and usage patterns that customers will actually use. The authors of the Web service specifications have always implicitly claimed to do this, but an examination of the rosters of the various standard authoring activities clearly reveals a lack of customer involvement. This inevitably leads to long protracted (and sometimes heated) discussions about what the specifications should, or should not, support. Disagreements over what the scenarios themselves should look like are also common. Having direct customer involvement in the discussion could go a long way towards reducing some of the tension and the length of the process.

### **Test scenarios based on customer requirements**

Unfortunately, interoperability scenarios often end up focusing on just what the most vocal Web service vendors can support. This results in scenarios that are based on

one or more vendors' code and not customer needs. The WSTF is trying to change this. The WSTF hopes to start with a business problem, apply the appropriate Web service specification, and then test—almost the opposite of what has happened in the past.

The goal of interoperability testing should not be to verify that scenarios can interoperate; it should be to verify that specifications actually meet customers' requirements in an interoperable fashion. Achieving this goal requires something that was missing from previous interoperability efforts: customers. The success of the WSTF will depend on whether true customer requirements are brought to the table in an objective way. Ideally, this would be done by having the customers themselves join the group and offer suggestions and guidance on the scenarios. As the WSTF is being announced, it already has several customers who are part of the group. The role of the customers is to offer ideas for scenarios and help keep the group focused on actual business requirements.

The number of scenarios currently being tested is small, but the shift in focus has already exposed several interoperability issues (as well as a serious design flaw) that were never noticed during previous interoperability testing. So even with all of the success of previous testing efforts, serious issues have been overlooked. Again, this has been due to the testing being focused on specific implementations of features, rather than on testing whether those features are actually usable and meet customers' needs. The result is that customers end up being the first ones to see these problems—this, of course, is not good for anyone.

## **The foundation of functional testing**

While being customer focused is critical, it is important to note that traditional *functional testing* also has its place in the big picture, as well as within the WSTF. If viewed as a set of building blocks, functional testing is the foundation on which all other testing will happen. This means that while the WSTF is trying to test at a higher level than what has been done in the past, it must also continue to ensure that the foundation of the Web service specifications themselves are sound.

In practical terms this means that most of the scenarios that have been tested in the past will gradually be brought into the WSTF. This will provide a transition from the previous testing efforts to WSTF. Vendors who participated in the previous interoperability events should be able to use the same code within the WSTF for these functional tests. This also will provide for regression testing.

## **Long lived endpoints**

Previous interoperability testing efforts were point in time statements. If successful, the test proved that a certain set of implementations could get together and interoperate once. There was no guarantee that things would continue to work properly in the future, nor was there a way for new implementations to verify that

they would interoperate with anyone else. Each implementation could (and probably does) set up other implementations *in house* to do this interoperability testing, but that is a very expensive and time consuming effort—an effort each vendor would need to duplicate. The WSTF tries to reduce this overhead by providing a shared, and long lived, set of testing endpoints that are available to the entire community.

The idea of *long lived* endpoints differentiates WSTF from other Web service testing efforts. Within the WSTF, each scenario will have a list of endpoints that support that scenario. By hosting and listing their endpoints, each implementation is agreeing not just that it supports the scenarios as described, but that it will try to keep those endpoints available for others to use for an indefinite amount of time. This means that the testing done within the WSTF is not a point in time statement; it's a continuous testing effort.

As vendors update products or develop new implementations, they can use the endpoints to verify that their new code works and ensure that bugs have not been introduced into existing implementations. While this could have been done in the past by each vendor hosting other vendors' implementations, the WSTF allows the true experts to host their own code—freeing developers to focus on their own implementations.

WSTF provides URLs that point to the endpoints themselves, not the actual implementations of the specifications. This means that vendors can test code that has not yet been shipped. They no longer need to wait until other vendors' products are publicly available before verifying that the latest versions have not introduced any regressions.

### **Forward looking scenarios**

In addition to the two testing views already discussed (customer centric and specification centric), WSTF also provides a third view: *forward looking* scenarios. Forward looking scenarios try to use the specifications in ways that have not been done in the past. This serves as an attempt to stay one step ahead of where actual deployments are so that, vendors see problems before their customers do.

Forward looking testing can be as simple as doing new compositional testing—testing the Web service specifications in combinations that have not been tested before. For example, WS-ReliableMessaging and WS-Transaction have both been available for several years, but there has never been any formal testing done where both are used together. There is no reason to expect they would not compose nicely, but until testing is actually done, it is not certain.

Another type of forward looking scenario is one that exposes a hole in the current Web service architecture. These scenarios would most likely start by describing a business requirement. Then while going through the process of trying to figure out which Web service specifications are best suited to solve that problem, the

community might realize that something new is needed.

### **Avoiding vendor lock-in**

Today when new needs related to Web service specifications arise, the solutions that are developed often end up being very vendor specific. The customer, working with a particular Web service vendor, might be able to develop a solution that meets their immediate needs, however, there is a very good chance that it will be tied to that one implementation. This means that the odds of the solution being interoperable (or even portable) drops dramatically. This ultimately defeats one of the main selling points of Web services—lack of vendor lock-in.

By providing a place for the Web services community to discuss ideas and concerns in a way that isn't provided elsewhere, the WSTF offers a way to help avoid the creation of solutions that are not interoperable. Most other forums for the Web services community are either focused on the development of a certain specification or on a particular implementation. By their very nature, the discussions within these forums are limited or skewed. Because the WSTF is community driven and open to everyone, discussions can go on there that might not otherwise happen.

For example, in the case of the vendor specific solution described earlier, if the discussion had instead happened within the WSTF, then more than one vendor could have participated. If there was a solution available that didn't require any kind of invention (a new Web service specification), then all of the vendors working together could ensure that the solution wouldn't lock customers into one particular product. If the problem could not be solved with existing specifications, the community could decide that a new Web service specification (or perhaps Profile, or change to existing specifications) is needed. Having participation from the entire community is preferable to a single vendor producing a one-off type of solution that locks the customer into one implementation choice.

The WSTF is not a specification development organization, but it does provide a venue for initial discussions to take place. As discussions progress, it will be up to the members of the WSTF (the Web services community at large) to decide when and what other appropriate actions should be taken.

## **Structure and governance**

The WSTF is actually a very simple and lightweight organization. At its core, the Web site consists of nothing more than a set of scenarios. The scenarios are managed by the members themselves in a manner similar to a traditional wiki site. Any member is welcome to modify any document at any time—similar to the open-source model.

### **The importance of privacy**

The one key difference between WSTF and the typical open-source model is the *private* nature of the WSTF. Within the open-source model, all work is visible to the entire world immediately. For some very important reasons, most of the work within the WSTF is done in private.

The most important reason for the private nature of the WSTF is due to the members' desire to have free and open discussions without the fear of any negative consequences. Members should be able to discuss new ideas, concerns, and even short-comings of products without worrying that their comments might later be used against them in some negative way. For this reason, the Participation Agreement that new members sign covers these two basic concepts (see the actual agreement for complete details):

- Anything said or done within the WSTF must remain private to the WSTF. The one exception to this rule is the publication of scenarios (more on this follows below).
- Anything brought to the group during the course of the discussions (for example, an idea for a scenario), is brought to the group under Royalty Free terms. This protects the members of the group from being sued by other members for patent infringement.

### **Publishing mature scenarios**

As mentioned above, scenarios are the only things visible to non-WSTF members. Once a scenario has reached a certain level of *maturity*, the implementers of that scenario can vote to allow non-WSTF members to see it—that is, they agree to publish it. A mature scenario is defined within the WSTF Charter as meeting these criteria:

- The WSTF members working on the scenario agree that it's ready for publication—how they decide this is up to them.
- The scenario has at least five different implementations.
- At least two thirds of those implementers agree to have the scenario published.

In many ways, this approval process is similar to how open-source communities vote on whether to release a new version of their code. The difference is that with WSTF, unpublished scenarios are not visible to non-members. The reason for this is that if the WSTF is successful and becomes a place where the entire Web services community jointly develops interoperability scenarios and tests them, then a little bit of a safety net is needed. The founders of the WSTF wanted to ensure that the organization remains truly community based and not dominated by one or two big vendors.



Requiring a scenario to reach a minimum set of requirements before it is published, provides a level of assurance that a *WSTF published scenario* has a broad community of support. If all scenarios were immediately available to everyone, there would be no way to know whether an individual scenario is a truly interoperable scenario and is supported by a large number of vendors, or whether it is simply one vendor's solution and could potentially lead to *lock-in*.

It's important to note that the publication of a scenario doesn't have much effect on the testing effort around that scenario. Published scenarios and unpublished scenarios have the same WSTF features available to them—the only difference is whether non-members can see them. This means that even if a scenario isn't supported by five or more members, it can still leverage all of the same benefits of the WSTF (the Web site, the mailing lists, etc.).

## Scenario artifacts

In addition to its detailed description and documentation, each scenario has three additional artifacts: supplemental files associated with the scenario document (for example, schema documents, diagrams, etc.), a mailing list specific to the scenario, and a list of endpoints.

In addition to the WSTF mailing lists for general topic discussions, each scenario provides its own unique mailing list. This allows members to limit the amount of mail they receive to only those scenarios that are of interest to them.

The *list of endpoints* associated with each scenario is similar to what has been provided with previous interoperability testing. Each implementation of a scenario provides a URL that points to an implementation of that scenario. Along with the URL, an endpoint optionally may provide additional information such as:

- Constraints about the implementation (for example, which version of SOAP it supports).
- Which version of a product is hosting the endpoint.
- Sample code. Members of the WSTF are encouraged to provide the implementation of their sample applications. Because the scenarios themselves will probably be written in a very generic fashion, the application code should not contain proprietary information. So providing the implementation could help jump-start customers who will want to mimic those scenarios in their own deployments.
- Additional documentation. Sometimes the issues customers run into when trying to deploy their code are not related to bugs in the SOAP stacks or in the specifications, but rather insufficient documentation around how to configure the products. For each scenario, vendors can include the exact

configuration information (or any other helpful data) to help customers reproduce the environment themselves.

- A contact in case there are questions about the endpoint.

The owner of each end point decides how much optional information to provide with the endpoint URL. The owner of each endpoint is also free to choose whether the endpoint appears on the external endpoint page after the scenario is published. This allows a vendor's participation in a certain scenario to remain private to the group. This is important because there have been times in the past when companies wanted to participate in the testing of a specification, but did not want it publicly known that they were supporting that specification.

Owners are generally encouraged to make their endpoints public when a scenario is published so that they can benefit from testing by non-WSTF members. However, a company may have a variety of reasons for wanting to remain private, and the WSTF provides this option in order to address this situation.

## Interoperability testing is key

Of course, the main purpose behind the WSTF is the interoperability testing itself. The actual testing will follow the same model that has been used many times in the past. Implementations will use the list of endpoints associated with a scenario to verify that the code on both sides works as expected.

If things go well during testing, the discussion around a scenario will probably be minimal. However, if things do not go as expected, then interested members will discuss their concerns through the scenario-specific mailing list. Often these discussions will generate *best practices* around how the specifications should be used or guidance on the possible choices application designers have. These will then typically make their way into the scenario documents under a Findings section.

As mentioned previously, discussions around a scenario could lead to ideas for new specifications or Profiles. In other cases, the discussions may lead to the exposure of issues with existing specifications. After any initial discussion concerning existing specifications takes place within the WSTF community, future discussion should then be moved to a more appropriate venue. The WSTF itself does not bring these issues to the appropriate place, it is up to the individual WSTF members to do so.

For example, suppose that an issue was found with the WS-ReliableMessaging specification. While some initial ideas of how to solve the issue may be discussed within the WSTF, the members of the WSTF that view the issue as important enough to take it forward are expected to take the issue to the specification authors—probably the OASIS WSRX Technical Committee. That technical committee would then decide how best to solve the problem.

Unlike other interoperability efforts, the WSTF is not prevented from examining any Web Service specification at any stage of its development. Other groups are designed in a way that prevents them from leaving the confines of their tightly scoped charters. The WSTF does not have these kinds of limitations. It can test a specification during, or even before, it is in a standards body.

For example, WS-I can only examine specifications that have exited a standards body. Often times this means that it is too late to address any serious issues found with the specification. By contrast, the WSTF could choose to go ahead and begin testing and providing feedback to the new W3C Working Group standardizing several specifications such as WS-MetadataExchange. The WSTF could begin such an effort long before the Working Group actually begins its own testing and well before the specification is complete. This can uncover obvious bugs in the specification as well as less obvious compositional issues that might only be found through bringing several Web service specifications together at once.

## It's all about participation

Because one of the goals of the WSTF is to be a truly open testing forum for the entire Web services community, the barrier to entry is kept very low. There are no membership dues or fees. The only requirement is that members sign the Participation Agreement. The Web site is relatively small and is being hosted by IBM, but it contains no IBM branding, logos, or marketing messages.

All members of the Web services community are encouraged to join WSTF. In addition to there being no fees, there is also no approval processes.

While it is hoped that all WSTF members participate to some degree, there is no minimum participation requirement. Members decide how much or how little they participate. Members who wish to simply join and read the mailing list are welcome to do so.

Participation in the development of scenarios is open to anyone at all—not just Web service vendors. Participation by non-vendors is actually critical to the WSTF's success. To ensure that real-world testing is actually being done, the WSTF will need customers, ISVs, and others to provide suggestions and guidance throughout the scenario development process.

## What's in the future for WSTF?

This article has used the term *Web services* as a synonym for SOAP, but the WSTF is actually not limited to just SOAP-based Web service testing. There is nothing that would prevent the WSTF to extend its testing into other *Web service* testing.

For example, the WSTF would allow, and even encourage, the testing of domain-specific uses of SOAP/Web service. Testing of REST/Web services would also be allowed and will likely take place in the not too distant future.

The WSTF isn't just about SOAP-based interoperability testing—it's about Web service interoperability testing, and the community itself will decide what that means over time.

## Conclusion

The WSTF was formed to focus on ensuring that issues with the interoperability of the WS-\* set of specifications are found and addressed before customers see them. The structure of the WSTF also provides for a much broader benefit to the Web services community at large. By providing a common testing ground and location for sharing ideas, the WSTF can fill some of the gaps in the community and open development aspects of Web services.

The WSTF can provide a forum where someone can speak to all types and sizes of players within the Web services community. The organizers of the WSTF have tried hard to ensure that the WSTF is focused on technical discussions with limited room for politics—hopefully, this will continue.

For more information see the WSTF web site at: <http://www.wstf.org> or send an e-mail to [admin@wstf.org](mailto:admin@wstf.org).

# Resources

## Learn

- In the [Architecture area on developerWorks](#), get the resources you need to advance your skills in the architecture arena.
- Browse the [technology bookstore](#) for books on these and other technical topics.

## Get products and technologies

- Download [IBM product evaluation versions](#) and get your hands on application development tools and middleware products from DB2®, Lotus®, Rational®, Tivoli®, and WebSphere®.

## Discuss

- Check out [developerWorks blogs](#) and get involved in the [developerWorks community](#).

# About the author

## Doug Davis

Doug Davis is an architect in the Software Standards division of IBM. His previous roles include technical lead of the Emerging Technologies Toolkit, WebSphere Machine Translation, Team Connection, and the IBM Fortran 90. He is currently a member of several industry organizations including WS-I, OASIS, DMTF and the W3C.

# Trademarks

IBM, the IBM logo, ibm.com, DB2, developerWorks, Lotus, Rational, Tivoli, and WebSphere are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. See the current list of [IBM trademarks](#).