

Web Services Composite Application Framework: Key Messages and FAQ

GENERAL

Q. What is the Web Service Composite Applications Framework (WS-CAF)?

A: WS-CAF is aimed at solving problems that arise when multiple Web services are used in combination, also known as *composite applications*.

WS-CAF includes three specifications:

- Web Service Context (WS-CTX), which models a Web services context data structure as a Web resource, accessible via standard URLs. WS-CTX is responsible for context management and defines a way for arbitrary services to augment the context.
- Web Service Coordination Framework (WS-CF), which defines a software agent called a *coordinator* that takes responsibility for augmenting the basic context and disseminating context information. Web services in a composite application register with coordinators to ensure message results are communicated between the coordinator and services in a reliable manner.
- Web Service-Transaction Management (WS-TXM), which defines three distinct transaction protocols that can be plugged into the coordination framework for interoperability across existing transaction managers, long running compensations, and business process automation. This is a “live” document, in that the authors intend for other transaction models to be added to it when the need arises.

Q. Why is WS-CAF needed?

A: Composite applications have unique requirements that are not yet addressed in a standard way, such as a method for sharing common information (context) and enabling the success or failure of individual Web services to be tied to the success or failure of a larger unit of work comprising multiple Web services. Thus, WS-CAF addresses some of the last remaining items to be standardized for Web services – a generic context mechanism, a context management service, and asynchronous multi-protocol transactions.

Shared context in a composite Web services application could include items such as security credentials -- so that a person could log in once and invoke multiple Web services without having to log in again -- a database connection over which to perform multiple operations from multiple Web services without having to establish a new connection each time, or a device address to which to post results from multiple Web services without having to search for the device’s network address each time. Shared context can also be viewed as context that needs to be “carried” across multiple operations.

Tying the success or failure of individual Web services into a larger unit of work means the ability to group multiple Web services into a single transaction with various, configurable properties to ensure that composite operations reliably produce a known state regardless of the failure of one or more Web services to successfully complete.

WS-CAF also supports multiple transaction protocols, including long running actions with compensations and asynchronous business process flows.

Q. Can you give an example of how WS-CAF would be used in a business scenario?

A: The ability to scope arbitrary units of distributed work by sharing common *context* is a requirement in a variety of distributed applications, such as choreography and business-to-business interactions. Imagine booking a holiday, where you want to contact a number of different flight shops, hotels and car rental organizations. You'll want to make sure that the work they do is associated with your work somehow. Scoping makes it possible for Web services participants to be able to determine unambiguously whether or not they are in the same composite application (the travel scenario) and what it means to share context.

A composite application is defined as a collection of Web services that executes in a specified sequence for the purpose of carrying out multiple operations on a shared resource, such as a database, display, or XML document, in which the unresolved failure of an individual Web service execution causes the failure of the entire application. The context can contain information such as security tokens or attributes, transaction identifiers, file ids, session ids, etc. This is because in the real world applications already exist which do this scoping in an ad hoc manner with these types of "context". What WS-CAF does is try to standardize how to construct and manage contexts and make the application developer and users life easier.

Using the basic context service, composite Web services can manage and interpret shared context themselves, or optionally, depend upon a shared Web service with which they each interact to manage and interpret the context. Using the Coordination Framework, Web services register with an external service that takes responsibility for managing and interpreting the shared context, and coordinating protocol-specific notification messages sent to participants upon completion of the shared work. Using one or more of the pluggable WS-TXM transaction models, the coordinator can coordinate completion activities depending upon the outcome of the shared work and the selected transaction model.

Q: Will this work be donated to a standards organization?

A: Yes, the work will be donated to a recognized standards organization and will be licensed under reciprocal royalty-free terms – no RAND terms at all. In order to participate in this work the authors have signed agreements that specify the work will be

royalty free and must go to a standards organization soon after release of the specifications.

Q: How can other companies get involved?

A: There have actually been a number of companies who have expressed a great deal of interest in this work while the initial versions of these specifications were being developed. Many of these companies are interested in participating in the evolution of the WS-CAF specifications once they are contributed to a standards body. We welcome and encourage other organizations interested in Web services standardization to join this effort in an open collaboration, and to bring their own ideas, requirements, and technologies (as long as they are offered on a royalty-free basis) to the table.

Q. Aren't the authors promoting specifications that duplicate existing work from Microsoft and IBM?

A. No. While these specifications may seem similar to the work published last August by IBM and Microsoft, there are some very important differences:

- The WS-CAF set of specifications addresses two major unresolved areas of Web services *standardization*:
 - Managing information (context) shared across multiple Web services composed into a larger unit of work such as processing a purchase order and executing a series of Web services within a transaction. Until now, there has been no common method for context management, and different working groups and vendors have been cobbling together their own specific (non-general) and non-interoperable solutions for managing and sharing this information. WS-CTX solves this problem by providing a flexible, standard mechanism to manage Web services context, improving interoperability, eliminating potential duplication of effort, and preventing other groups from being distracted from their primary goals.
 - Web service compositions and transactions. WS-CAF provides a complete and open framework to support Web service transactions – for example, a major innovation in WS-TXM involves federating different transaction models, so you can bridge between, for example, MQ Series and EJBs)
- The WS-CAF specifications are on track to become true standards, with a known development and review cycle, royalty-free licensing terms and participation open to all interested parties.

It is also important to note that WS-CAF was specifically designed as an open, 'pluggable' framework; so for example, WS-Transaction is supported as one of the transaction protocols that can be used in conjunction with WS-CAF.

Q: Why do the authors care about open standards?

A: We care about open standards because our customers demand them and because we feel true open standards are necessary for the industry and the economy to progress.

For customers, the promises of open standards are predictability, interoperability and reduced costs. By "open," we mean specifications that are reliable, free from legal encumbrances (truly royalty-free licensing terms), able to work across development and deployment environments, subject to peer review and input throughout their lifecycle, and aligned with general industry and customer needs. By "standard," we mean specifications that are developed in recognized standards-setting organizations.

Q: What is the benefit of royalty-free licensing?

A: While much of this may be hidden to end users (except for possible added cost), there is a large community of businesses and users that actually use these specifications and standards directly to build their products and services. Particularly with Web services, which is envisioned as enabling a large and rapidly growing marketplace/ecosystem of services, it is critical to prevent the foundational standards to be threatened by IP that is not publicized.

Even end-user customers and businesses who don't build the services themselves are concerned with this when they consider the long-term impact of having a key foundation for their business controlled by a couple of big players. Open and royalty-free standards provide predictability, interoperability and reduced costs. How many of today's successful e-businesses would be content to discover that a company who once had engineers contribute, for example, to the TCP/IP or HTTP specification efforts, now claims intellectual property rights over their contributions, and demands royalties for use of those standards?

Q: When and how do you plan to tie/incorporate this work into current specs like BPEL, WSCI, WS-Sec, GXA specs, etc.?

A: Now that these specifications have been published, we plan to work closely with recognized standards working groups to encourage the adoption of this common framework for composite applications in order to rationalize the Web services architecture and provide a standard, interoperable underpinning for Web services context. The Web Services Composite Application Framework was specifically designed to make it easy for other specifications and technologies such as these to take advantage of its facilities with minimal changes.

TECHNICAL

Q: What's the relationship of WS-TXM to the Business Transaction Protocol (BTP) work at OASIS?

A: WS-TXM supports interoperability across multiple transaction protocols, including BTP and other proprietary protocols, such as WS-Transaction (MSFT and IBM).

Q: What are the technical differences between WS-CAF and Web Service-Coordination (WS-C) and Web Service-Transaction (WS-T), which were published by MSFT and IBM in August 2002?

A: WS-CAF is compatible with WS-T and WS-C. The WS-CTX and WS-CF specifications in particular are a superset of WS-C that extends and improves upon WS-C, while WS-T is supported among the pluggable transaction protocols that can be used in conjunction with the WS-CTX specification.

The WS-CAF set of specifications enhances the functionality of WS-C and WS-T to provide:

- A separate, standalone generic context management mechanism (WS-CTX)
- Interoperability across multiple transaction protocols (e.g. WS-T is not required, as it is for MSFT and IBM's work)
- Innovate and unique support for long running business process flows, including occasionally connected computing devices

Comparison of WS-CAF (Web Services Composite Application Framework) and WS-CTX (Web Service Context) to IBM/MS' WS-C (Web Service Coordination):

First, WS-CTX is interoperable. It was designed to work with existing as well as new and evolving Web Services specifications that are in open development in standards organizations, such as specifications for choreography, security, and transactions.

WS-CTX takes the approach that context should be a first-class entity in the Web services architecture. As such, it is represented by a URI and can either be passed by value directly in operation calls or by reference. This allows contexts to be more easily managed in a large-scale environment.

MSFT and IBM's WS-C (Web Service Coordination) has no equivalent for WS-CTX (Web Service Context). WS-C's notion of context is tightly tied to the coordinator that created it, making it less flexible and less lightweight than WS-CTX. Context is more fundamental than coordination, and therefore benefits from a separate specification.

It is also impossible for multiple services to cooperate through WS-C's definition of "context." For example, if an application requires security, transactions and replication contexts, they would have to be handled in an unspecified, ad hoc manner. On the other hand, WS-CTX provides a lightweight structuring mechanism for any number of service types to participate within the lifecycle of the same activity (the definition of which is left up to the collection of services).

WS-CAF is an improved superset of WS-C. As with WS-C, WS-CAF is a generic framework for coordination. However, only WS-CAF has been architected to allow interoperable and portable implementations. It does not dictate implementation style. It

builds upon WS-CTX and can more easily tie into existing backend coordination infrastructures than WS-C.

The developers of WS-CAF have attempted to cover all aspects necessary to write interoperable coordination protocols rather than leave it up to individual developers, as WS-C does. The latter approach results in coordination protocols that cannot work across heterogeneous implementations, or even across different versions of the same protocol. The schemas used by WS-CAF have been designed to allow protocols to be specified precisely.

As with WS-C, WS-CAF has been designed to allow existing coordination infrastructures to be leveraged. However, only WS-CAF provides its own interoperable coordination protocol for implementers to use. This can be used, for example, to simplify protocol bridging between existing systems, reducing costs of implementation and time to market.

Comparison of WS-TXM (Web Service Transaction Management) to WS-T (Web Service Transaction):

WS-TXM's transaction model supports interoperability across multiple transaction protocols, including BTP, WS-TXM itself, and other proprietary protocols such as WS-Transaction. Unlike WS-TXM, WS-T *requires* use of WS-T and thus does not support interoperability across multiple transaction protocols. This is an additional cost for the customer.

Still, note that because transaction protocols are pluggable with WS-CAF, WS-Transaction can be used, if so desired.