**XML Web Services Infrastructure**

**.NET Framework 4**

[Other Versions](javascript:;)

http://i.msdn.microsoft.com/Areas/Epx/Content/Images/ImageSprite.png

* [.NET Framework 1.1](http://msdn.microsoft.com/en-us/library/sd5s0c6d(d=printer,v=vs.71).aspx)
* [.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/sd5s0c6d(d=printer,v=vs.80).aspx)

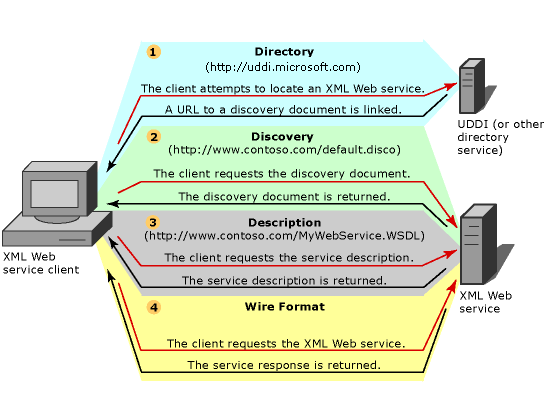
**This topic is specific to a legacy technology. XML Web services and XML Web service clients should now be created using** [Windows Communication Foundation](http://go.microsoft.com/fwlink/?LinkID=127777) .

XML Web services must be agnostic regarding the choice of operating system, object model, and programming language to succeed in the dissimilarity of the Web. Also, for XML Web services to benefit from the same widespread adoption as other Web-based technologies, they must be:

* Loosely Coupled: Two systems are considered loosely coupled if the only mandate imposed on both systems is to understand the aforementioned self-describing, text-based messages. Tightly coupled systems, on the other hand, impose a significant amount of customized overhead to enable communication and require a greater understanding between the systems.
* Ubiquitous Communication: It is unlikely that anyone builds an operating system now or in the near future that does not incorporate the ability to connect to the Internet, therefore providing a ubiquitous communication channel. As such, the ability to connect almost any system or device to the Internet ensures such systems and devices are universally available to any other system or device connected to the Internet.
* Universal Data Format: By adopting existing, open standards over proprietary, closed-loop communication methods, any system that supports the same open standards is capable of understanding XML Web services. Utilizing self-describing, text-based messages that XML Web services and their clients can share without knowing what constitutes each underlying system enables communication between autonomous and different systems. XML Web services achieve this capability using XML.

XML Web services employ an infrastructure that provides the following: a discovery mechanism to locate XML Web services, a service description for defining how to use those services, and standard wire formats with which to communicate. The following illustration shows an example of this infrastructure.

**XML Web services infrastructure**



|  |  |
| --- | --- |
| **Infrastructure piece** | **Role** |
| [XML Web Services Directories](http://msdn.microsoft.com/en-us/library/7e29kfs9(v=vs.100).aspx) | XML Web services directories provide a central location to locate XML Web services provided by other organizations. XML Web services directories such as a UDDI registry fulfill this role. XML Web service clients might or might not reference an XML Web service's directory. |
| [XML Web Service Discovery](http://msdn.microsoft.com/en-us/library/fxx6cfx2(v=vs.100).aspx) | XML Web service discovery is the process of locating, or discovering, one or more related documents that describe a particular XML Web service using the Web Services Description Language (WSDL). The DISCO specification defines an algorithm for locating service descriptions. If XML Web service clients know the location of the service description, they can bypass the discovery process. |
| [XML Web Service Description](http://msdn.microsoft.com/en-us/library/77axffs8(v=vs.100).aspx) | To understand how to interact with a particular XML Web service, it is necessary to provide a service description that defines what interactions the XML Web service supports. XML Web service clients must know how to interact with an XML Web service before they can use it. |
| [XML Web Service Wire Formats](http://msdn.microsoft.com/en-us/library/a1tx28sw(v=vs.100).aspx) | To enable universal communication, XML Web services communicate using open wire formats, which are protocols understandable by any system capable of supporting the most common Web standards. SOAP is the key protocol for XML Web service communication. |

**XML Web Services Overview**

**.NET Framework 4**

[Other Versions](javascript:;)



* [.NET Framework 1.1](http://msdn.microsoft.com/en-us/library/w9fdtx28(d=printer,v=vs.71).aspx)
* [.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/w9fdtx28(d=printer,v=vs.80).aspx)

**This topic is specific to a legacy technology. XML Web services and XML Web service clients should now be created using** [Windows Communication Foundation](http://go.microsoft.com/fwlink/?LinkID=127777) .

An XML Web service is a programmable entity that provides a particular element of functionality, such as application logic, and is accessible to any number of potentially disparate systems using ubiquitous Internet standards, such as XML and HTTP. XML Web services depend heavily upon the broad acceptance of XML and other Internet standards to create an infrastructure that supports application interoperability at a level that solves many of the problems that previously hindered such attempts.

An XML Web service can be used internally by a single application or exposed externally over the Internet for use by any number of applications. Because it is accessible through a standard interface, an XML Web service allows heterogeneous systems to work together as a single web of computation.

Instead of pursuing the generic capabilities of code portability, XML Web services provide a viable solution for enabling data and system interoperability. XML Web services use XML-based messaging as a fundamental means of data communication to help bridge the differences that exist between systems that use incongruent component models, operating systems, and programming languages. Developers can create applications that weave together XML Web services from a variety of sources in much the same way that developers traditionally use components when creating a distributed application.

One of the core characteristics of an XML Web service is the high degree of abstraction that exists between the implementation and the consumption of a service. By using XML-based messaging as the mechanism by which the service is created and accessed, both the XML Web service client and the XML Web service provider are freed from needing any knowledge of each other beyond inputs, outputs, and location.

XML Web services are enabling a new era of distributed application development. It is no longer a matter of object model wars or programming language beauty contests. When systems are tightly coupled using proprietary infrastructures, this is done at the expense of application interoperability. XML Web services deliver interoperability on an entirely new level that negates such counterproductive rivalries. As the next revolutionary advancement of the Internet, XML Web services will become the fundamental structure that links together all computing devices.

# XML Web Service Scenarios

**.NET Framework 4**

[Other Versions](javascript:;)



* [.NET Framework 1.1](http://msdn.microsoft.com/en-us/library/c8f1ys8a(d=printer,v=vs.71).aspx)
* [.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/c8f1ys8a(d=printer,v=vs.80).aspx)

**This topic is specific to a legacy technology. XML Web services and XML Web service clients should now be created using** [Windows Communication Foundation](http://go.microsoft.com/fwlink/?LinkID=127777) .

To better understand the valuable contribution of XML Web services, it is helpful to examine a few scenarios in which XML Web services could play a valuable role.

* [Simple Services](http://msdn.microsoft.com/en-us/library/6exz06fz(v=vs.100).aspx)
* [Application Integration](http://msdn.microsoft.com/en-us/library/ftcdxbs0(v=vs.100).aspx)
* [Workflow Solutions](http://msdn.microsoft.com/en-us/library/w3dsh02f(v=vs.100).aspx)

# Simple Services

**.NET Framework 4**

[Other Versions](javascript:;)



* [.NET Framework 1.1](http://msdn.microsoft.com/en-us/library/6exz06fz(d=printer,v=vs.71).aspx)
* [.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/6exz06fz(d=printer,v=vs.80).aspx)

**This topic is specific to a legacy technology. XML Web services and XML Web service clients should now be created using** [Windows Communication Foundation](http://go.microsoft.com/fwlink/?LinkID=127777) .

The most basic scenario fulfilled by XML Web services is that of providing some fundamental piece of functionality for its clients to use. For example, a challenge faced by e-commerce applications is the need to calculate charges for an assortment of shipping options. Such applications would require current shipping cost tables from each shipping company to use in these calculations.

Alternatively, an application could send a simple XML-based message over the Internet, using a standard transport protocol such as HTTP, to the shipper's cost calculation XML Web service. The message might provide the weight and dimensions of the package, ship-from and ship-to locations, and other parameters, such as class of service. The shipper's XML Web service would then calculate the shipping charge using the latest cost table, and return this amount, in a simple XML-based response message, to the calling application for use in calculating the total charge to the customer.

# Application Integration

**.NET Framework 4**

[Other Versions](javascript:;)



* [.NET Framework 1.1](http://msdn.microsoft.com/en-us/library/ftcdxbs0(d=printer,v=vs.71).aspx)
* [.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/ftcdxbs0(d=printer,v=vs.80).aspx)

**This topic is specific to a legacy technology. XML Web services and XML Web service clients should now be created using** [Windows Communication Foundation](http://go.microsoft.com/fwlink/?LinkID=127777) .

You can use XML Web services in a composite manner to integrate a seemingly disparate group of existing applications. The wide adoption of custom software throughout virtually every department of most companies has resulted in a vast array of useful but isolated islands of data and business logic. Due to the varied circumstances under which each was developed, and the ever-evolving nature of technology, it is a daunting task to create a functional grouping from these applications.

With XML Web services, it is possible to expose the functionality and data of each existing application as an XML Web service. You can then create a composite application that uses this collection of XML Web services to enable interoperability between the constituent applications.

# Workflow Solutions

**.NET Framework 4**

[Other Versions](javascript:;)



* [.NET Framework 1.1](http://msdn.microsoft.com/en-us/library/w3dsh02f(d=printer,v=vs.71).aspx)
* [.NET Framework 2.0](http://msdn.microsoft.com/en-us/library/w3dsh02f(d=printer,v=vs.80).aspx)

**This topic is specific to a legacy technology. XML Web services and XML Web service clients should now be created using** [Windows Communication Foundation](http://go.microsoft.com/fwlink/?LinkID=127777) .

XML Web services enable a powerful mechanism by which applications that constitute end-to-end workflow solutions can be created. Such solutions are appropriate for long-running scenarios such as those found in business-to-business transactions.

The BizTalk Framework provides an additional protocol layer that defines mechanisms to identify and address messages, define their lifetime, package them with attachments, deliver them reliably to their destinations, and secure their contents for authentication, integrity, and privacy.

Microsoft® BizTalk™ Server provides the infrastructure and tools for rules-based business document routing, transformation, and tracking infrastructure. The infrastructure enables companies to integrate, manage, and automate business processes by exchanging business documents (for example, purchase orders and invoices) among applications within or across organizational boundaries.

BizTalk Orchestration is a technology that is included with BizTalk Server for defining the behavior of a single XML Web service, as well as the composition of many XML Web services in building multi-party business processes.