BizTalk Framework 2.0 Draft: Document and Message Specification

Microsoft Corporation
June 2000

Draft Specification

Summary: This draft specification provides a general overview of the BizTalk Framework 2.0 conceptual architecture, including the BizTalk Document and BizTalk Message. It provides detailed specifications for the construction of BizTalk Documents and Messages, and their secure transport over a number of Internet-standard transport and transfer protocols.

Contents

1. Introduction

The growing maturity of Internet-based secure transport protocols, combined with ubiquitous support for these protocols across networking, hardware, and software platforms, is enabling businesses to develop new ways to facilitate efficient and automated interactions. These interactions can occur between their own internal lines of business; productivity and knowledge management applications; the applications used by their customers and partners; and services provided by their commercial and corporate providers.

The challenges associated with enabling such efficient, automated interactions between applications across business boundaries, and in a cost effective manner, are similar to those associated with enabling them within an enterprise or departmental boundary. However, a new dimension of challenges in the areas of security and reliability must be addressed in order to communicate with other organizations.

These challenges of interaction across business boundaries include, but are not limited to, the following:

- Lack of a sufficiently-flexible and rich universal language to specify, package, publish, and exchange both structured and unstructured information across application or business boundaries.
- Lack of a flexible and rich universal language to specify, package, publish, and execute transformation rules to convert information from one format to the other as application and business boundaries are crossed.
- Lack of middleware-neutral, application-level communication protocols that enable automated interactions across application or business boundaries.

Extensible Markup Language (XML) and XML-based schema languages provide a strong set of technologies with a low barrier to entry. These languages enable one to describe and exchange structured information between collaborating applications or business partners in a platform- and middleware-neutral manner.

As a result, domain-specific standards bodies and industry initiatives have started to adopt XML and XML-based schema languages to specify both their vocabularies and content models. These schemas are becoming widely published and implemented to facilitate communication between both applications and businesses. Wide support of XML has also resulted in independent solution providers developing solutions that enable the exchange of XML-based information with other third-party or custom-developed applications. Several solution- or middleware/platform-specific approaches have been taken to address the lack of middleware-neutral, application-level communication protocols. However, no single proprietary solution or middleware platform meets all the needs of a complex deployment environment.

These proprietary initiatives have generally resulted in customers facing broad interoperability issues on their own. The BizTalk™ Framework addresses these interoperability challenges in a platform- and technology-neutral manner. It provides specifications for the design and development of XML-based messaging solutions for communication between applications and organizations. This specification builds upon standard and emerging Internet technologies such as Hypertext Transfer Protocol (HTTP), Multipurpose Internet Mail Extensions (MIME), Extensible Markup Language (XML), and Simple Object Access Protocol (SOAP). Subsequent versions of the BizTalk Framework will be enhanced to leverage additional XML and Internet-related, messaging-standards work as appropriate.

It is important to note that the BizTalk™ Framework does not attempt to address all aspects of business-to-business electronic commerce. For instance, it does not deal directly with legal issues, agreements regarding arbitration, and recovery from catastrophic failures, nor does it specify specific business processes such as those for purchasing or securities trading. The BizTalk™ Framework provides a set of basic mechanisms required for most business-to-business electronic exchanges. It is expected that other specifications and standards, consistent with the BizTalk™ Framework, will be developed for the application- and domain-specific aspects.

2. Specification Scope and Evolution

This specification provides a general overview of the BizTalk Framework conceptual architecture, including the fundamental notions of BizTalk Document and BizTalk Message. It then provides detailed specifications for the construction of BizTalk Documents and Messages, and their secure transport over a number of Internet-standard transport and transfer protocols, as described below.

BizTalk Documents follow a number of rules for structure and content in order to provide rich functionality and predictable semantics. This specification describes the following aspects of BizTalk Documents and their semantics:

- Overall structure of BizTalk Documents.
- BizTalk headers for document routing, properties, catalog, and process management.
- Structure and handling of BizTalk Documents that require reliable delivery.

When implementing solutions using the BizTalk Framework, specific transport, encoding, and security mechanisms must be used to secure and deliver messages. This specification describes the following mechanisms and aspects of BizTalk Message encoding and transport:

- Transport bindings for Internet protocols (HTTP only; Simple Mail Transfer Protocol (SMTP), and File Transfer Protocol (FTP) to be added).
- MIME-based transfer encoding and attachment packaging.
- Signatures and encryption based on S/MIME and Public-Key Cryptography System (PKCS) (to be added).

This specification is intended to define messaging interaction between BizTalk Framework 2.0 Compliant servers, referred to as *BFC servers* in this specification.

2.1 Relationship to BizTalk Framework 1.0

The BizTalk Framework 2.0 specification is a major revision of the BizTalk Framework 1.0 specification. BizTalk Framework 2.0 includes the following new features:

- Transport bindings (HTTP only; SMTP, and FTP to be added).
- Reliable message delivery.
- MIME encoding for attachments.
- Security based on S/MIME and PKCS (to be added).

In addition, BizTalk Framework 2.0 has been influenced by many recent standards efforts including, but not limited to, the following:

- SOAP Version 1.1
- XML Schema Part 1: Structures
- XML Schema Part 2: Data types
- XML-Signature Syntax and Processing

The influence of SOAP 1.1 is most pervasive since BizTalk Framework 2.0 is an extension of SOAP 1.1, whereas BizTalk Framework 1.0 was "pre-SOAP." In addition, the opportunity for a major revision was used to rationalize the semantics, naming, and structure of many BizTags in the light of experience and the requirements of the new features in this specification.

One of the goals of BizTalk Framework 2.0 is to sufficiently explain wire-level behavior so that it's useful as the basis for interoperation among compliant servers. The semantics of many BizTags has been defined much more specifically than in BizTalk Framework 1.0. The structure and content of BizTalk Documents described in BizTalk Framework 1.0 have been

preserved wherever possible, but precise semantics and consistency with standards have been given higher priority in order to provide a solid foundation for the future.

2.2 Versioning Model

BizTalk Framework 2.0 follows SOAP 1.1 in not defining a traditional versioning model based on major and minor version numbers. The version is implied by the namespace URIs used to qualify the BizTalk-specific header entries defined in this specification. Normal SOAP 1.1 rules for the **SOAP-ENV:mustUnderstand** attribute imply that if the header entries that are required to be understood carry the wrong namespace or are deemed ill-formed in some other fashion, the BFC server should respond with a **SOAP-ENV:mustUnderstand** fault.

In the context of the HTTP binding specified in section 10, this fault indication may be returned in the HTTP response. However, if the message is processed asynchronously, the HTTP response will be **202 accepted** and the fault should be returned asynchronously, whenever possible. See section 10 for further discussion of transport protocol binding.

3. Dependencies

3.1 Normative Specifications

Each BizTalk Framework document lists the existing or emerging Internet standards that it is built upon as normative references. Some of the content of the normative references may need to be reproduced for expository purposes in BizTalk Framework specifications. In all such cases, the normative references are authoritative. Every effort has been made to avoid discrepancies between the normative references and their usage in BizTalk Framework specifications. However, if a discrepancy is found, the normative reference provides the correct interpretation and the BizTalk Framework specification is in need of correction.

The following specifications are normative for this specification:

- Extensible Markup Language (XML) 1.0
- Simple Object Access Protocol (SOAP) Version 1.1
- Namespaces in XML
- Uniform Resource Identifiers (URI): Generic Syntax
- ISO 8601: Representations of dates and times
- Hypertext Transfer Protocol—HTTP/1.1
- XML Media Types
- MIME Part One: Format of Internet Message Bodies
- MIME Part Two: Media Types
- MIME Part Three: Message Header Extensions for Non-ASCII Text
- MIME Part Four: Registration Procedures
- Content-ID and Message-ID Uniform Resource Locators

3.2 Non-Normative Specifications

The following specifications have had an influence on this specification, but the relationship is not foundational and their content is not normative for this specification:

- XML-Data Reduced (XDR)
- XML Schema Part 1: Structures
- XML Schema Part 2: Data types
- XML-Signature Syntax and Processing

3.3 Use of XML Schema Data Types

This specification uses the type-qualification **xsi:type** attribute as well as a number of specific data types from the XML Schema specifications. These are listed below with explanations. This specification, however, does not mandate the use of a specific method for defining XML schemas.

The **xsi:type** attribute allows an element to explicitly assert its type in a specific XML document instance. This can be used to validate the structure of the element.

The data type **timeInstant** represents a specific instant of time. The value space of **timeInstant** is the space of combinations of date and time of day values as defined in section 5.4 of the ISO 8601 standard.

The **uriReference** data type represents a URI reference as defined in Section 4 of Request for Comments (RFC) 2396. A URI reference may be absolute or relative, and may have an optional fragment identifier.

A **complexType** is an element with content that is not a simple type, such as a string or a decimal number; the element contains subelements and/or attributes with their own content.

4. BizTalk Concepts

4.1 Terminology

This document uses a set of BizTalk-specific terms, as defined below:

- **BizTalk Framework Compliant (BFC) Server.** A BFC Server is represented by the set of services providing the message-processing functionality defined in the BizTalk Framework specifications.
- **Application.** An Application is the line-of-business system where the business data or logic are stored and executed. An application also includes any additional adapters that may be required to emit or consume Business Documents (see below) and communicate with a BFC server.
- **Business Document.** A Business Document is a well-formed XML document containing business-transaction data. This transaction data may represent a purchase order, invoice, sales forecast, or any other business information. One or more Business Documents form the body of a BizTalk Document (see below).

  The BizTalk Framework does not prescribe the content or structure (schema) of individual Business Documents. The details of the Business Document content and structure, or Schema, are defined and agreed upon by the solution implementers.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.