IBM WebSphere Business Connection

Architecture Overview

Version 1.1.0

Note!

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 9.

First Edition (September 2002)

This edition applies to Version 1, Release 1, Modification 0, of *IBM® WebSphere® Business Connection* (5724-D26) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. You can send them to the following address:

IBM Canada Ltd. Laboratory
Information Development
8200 Warden Avenue
Markham, Ontario, Canada L6G 1C7

Include the title and order number of this book, and the page number or topic related to your comment.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Architecture Overview

The IBM(R) WebSphere(R) Business Connection offerings enable business-to-business process integration and data sharing among trading partners of all types and sizes. Business Connection offerings provide a scalable set of editions that link businesses using industry-standard and Web-services protocols. Companies can start with a simple, low-cost Web-services connection siness Connection Express, and scale to support additional partners and more complex business processes with Business Connection and Business Connection Enterprise.

Enterprises and their partners benefit from reduced integration costs and faster deployment of new processes and services using open standards and proven technology.

This document describes the architecture of the IBM WebSphere Business Connection offerings and how processes and data are exchanged among companies and their trading partners.

In a typical scenario, such as the one shown below, the Business Connection editions might be linked to a Business Integration hub implementation (BI hub), which is a reference architecture and implementation that can be built through a services contract with IBM.

Figure: Business Connection editions linked to a BI hub implementation

See your IBM marketing representative for information on configuring a BI hub.

The architecture

The architecture on which Business Connection is built is discussed in this section. The architecture can be thought of as a series of layers, with each layer representing a set of functions.

The private processes and applications of an enterprise (depicted in the Private Process Engine block) are shown in the architectural stack even though they are not part of the Business Connection offering. They play an integral part, however, because a key goal of any B2B exchange is to interact with the private processes of a trading partner. The other layers of the stack move messages (requests and responses, for example) in and out of the Private Process Engine so that an enterprise can share processes and data with its trading partners.

Figure: The Business Connection architecture layers, with the Public Process Engine communicating with the Private Process Engine

The Business Connection Enterprise Edition supports a variety of communication protocols; however, regardless of which communication protocols or which adapter services are used, the architecture remains the same.

1. The B2B Communication Handling layer receives requests and sends responses (in the form of messages) to and from external trading partners. This layer is the entry point into and exit point out of a Business Connection system.

2. The Adapter Services layer transforms the business message associated with the request into a representation that can be handled by the processes deployed in the Public Process Engine.
3. The Public Process Engine choreographs the services that are available in the Private Process Engine with those of its external partner systems.
4. The Tools and Services layer traverses the architecture and includes useful processes, utilities, and samples.

The following sections provide details of each of the layers of the architecture.

B2B Communication Handling layer

Requests come in through the B2B Communication Handling layer and are transformed and transported upward through the layers until the requested interaction takes place.

Figure: Communication across enterprises through the layers of the stack

The B2B Communication Handling layer functions include authentication and authorization checking, data validation, and transport handling.

The B2B Communications Layer has a series of transport listeners that wait for incoming requests (in the form of business messages) to arrive at the enterprise. The Business Connection Enterprise Edition supports a variety of protocols, and if a message arrives that conforms to one of the protocols, the B2B Communications Handling Layer receives it. The transports with which this layer can interact are:
- Web services
- Electronic Data Interchange over a Value-Added Network (EDI-VAN)
- Electronic Data Interchange over the Internet (EDI-INT)
- RosettaNet Implementation Framework (RNIF)

Web services are handled by the Web Services Gateway, which is a lightweight proxy gateway for simple Web-service interactions between companies (Simple Object Access Protocol or SOAP operations). This gateway allows internally defined services to be exposed to external systems and allows externally defined services to be accessed.

The Web Services Gateway, as shown in the following illustration, includes several components that perform services as the business request flows through the gateway.

Figure: How requests flow through the Web Services Gateway and into the SAI

The transport handler:
- Verifies the general message integrity
- Checks the content type and length
- Checks the document encoding
- Handles transport de-enveloping

The transport handler passes the message to the protocol handler, which:
- Parses the message
- Validates the structure, sequence, attributes and schema of each part of the message

- Sends an acknowledgement to the remote trading partner
- Interacts with the channel services to perform functions such as timeout tracking

The transport handler sends the message to the Web Services Gateway Manager, which provides authorization and routing to the appropriate collaboration. The message is eventually sent to the Server Access Interface at the Adapter Services Layer.

Adapter Services layer

The Adapter Services layer transform**s** the business documents associated with the requests that flow in through the B2B Communication Handling layer. In other words, the Adapter Services layer acts as a bridge between the B2B Communication Handling layer and the Public Process Engine layer, transforming requests into a format that the Public Process Engine can recognize.

The Adapter Services Layer functions are performed by CrossWorlds(R) connectors and data handlers or by CrossWorlds Server Access Interface.

- Inbound requests through the Web Services Gateway are handled by the Server Access Interface (SAI)
- Outbound Web-service calls from a collaboration are handled by the SOAP connector
- RosettaNet requests come through the TPI server
- EDI-INT requests come through the TPI server, are routed to the WebSphere Data Interchange (DI) server, and are then handled by the MQ connector
- EDI-VAN requests, which come through the DI server, are handled by the MQ connector

Figure: How requests flow through different gateways and servers

The connectors and data handlers transform the data received from the B2B Communication Handling layer (which is in a serialized form) into *business objects*. The data handler parses the data and populates the attribute values for the business object.

A connector is made up of a connector agent, which interacts with the B2B Communications Handling layer, and a connector controller, which interacts with the public process engine (the InterChange Server). The output of the connector agent and data handlers is an application-specific business object.

Public Process Engine layer

Collaborations execute within the Public Process Engine (the InterChange Server). Because collaborations are application-neutral, they work with generic business objects. The connector controller therefore maps the application-specific business object into a generic business object for use by the collaboration, as shown in the following illustration:

Figure: The connector controllers map business requests

A Web service that comes in through the Web Services Gateway interacts with the Server Access Interface rather than a controller.

Figure: The SAI transforms Web service requests

Collaboration end points (called *ports*) can be “bound” to

- Other collaborations
- Connectors
- External call requests (through the Service Access Interface)

One of the ports, the trigger port, identifies which business object or call will start the collaboration. In the case of the incoming Web services request, the Service Access Interface calls the collaboration (and sends the business object) to start the collaboration.

The collaboration processes the business object and sends the resulting data request to an application connector (to which a port has been bound) for the private application. The connector paradigm described earlier is repeated again; in this case, the generic business object is transformed to an application-specific business object.

Figure: How business objects pass from collaborations to private applications

The response data from the application is mapped back to a request business object and sent back to the collaboration. The collaboration returns the requested data or results of requested action to the Server Access Interface or to the appropriate connector in business-object format. Eventually, the business object is transformed into the format required by the appropriate server or gateway and is returned to the trading partner.

Tools and Services layer

The Tools and Services layer includes:

- System management services, which let you monitor and control your entire system from one console
- Solution management services, which let you monitor processes at the solution level
- A registration and provisioning service, allowing you to provide easy, efficient sign-up of your trading partners
- A document exchange service, allowing you to reliably exchange even very large documents with trading partners

  The four services listed above are described in Administering the System.

- Security services that help you control access to your system, as described in Installation and Configuration
- Web-services samples, which provide guidance on how to enable Web services, as described in the Web Services set of documents

---

How requests flow through the system

This section describes how various requests (messages) flow through the system. It shows how a request from a trading partner flows in through the Web Services Gateway and how a request flows out of the Web Services Gateway. It also describes how a request from a trading partner flows through the TPI Server.

Web Services Gateway flow - inbound

This section describes how a request (in the form of a SOAP message) from a trading partner flows through the Business Connection system of an enterprise and interacts with a private application of that enterprise.

1. The Web Services Gateway makes its Web services known to the trading partner via an industry-standard Web Service Description Language (WSDL) file, and the trading partner uses that information to send a SOAP message to the enterprise.
2. Inside the Web Services Gateway of the enterprise, a transport handler verifies the general message integrity.
3. The protocol handler:
   a. Validates the structure, sequence, attributes, and schema of each part of the message.
   b. Sends an acknowledgment to the remote trading partner.
   c. Uses channel services to track timeout periods and retry counts.
   d. Passes the message to the Web Services Gateway Manager.
4. The Web Services Gateway Manager:
   a. Provides authorization checking.
   b. Provides routing to the target service.
5. The Web Services Gateway Manager passes the message to the enterprise Web service, which invokes the CrossWorlds Server Access Interface.
6. The Server Access Interface delegates the request for data conversion to a data handler. (The incoming business message has to be converted to the generic business format expected by the collaboration.)
7. The Server Access Interface invokes the collaboration specified by the trading partner, using the converted business object as input.
8. The collaboration performs the function in the business-process flow, passing the business object to the appropriate application connector.
9. The application connector controller invokes a mapping function to transform the generic business object into an application-specific business object.
10. The application connector controller sends the application-specific business object to the application connector agent, which interacts with the application. The application connector agent uses application-specific information to drive its interactions with the application.
11. The response data from the application is mapped back to the request business object and sent back to the collaboration via the connector.
12. The collaboration sends the requested data back through the Server Access Interface.
13. The Server Access Interface returns the data in the original protocol format to the Web service, which returns the data to the Web Services Gateway.
14. The Web Services Gateway returns the information to the trading partner.

Web Services Gateway flow - outbound

This section describes how a request for an external service flows out of a system.

1. A triggering event causes an outbound collaboration to be activated. For example, the request to send a document might be a triggering event that causes its associated connector to invoke a collaboration.
2. The connector passes a business object to the collaboration.
3. The collaboration accepts the business object from the connector (its *From* port).

4. The collaboration delivers the message to the SOAP connector controller, which converts (maps) the generic business object into a SOAP application-specific business object.
5. The SOAP application-specific business object is converted by the SOAP connector agent to a SOAP message and sent to the internal Web service, which is the Web Services Gateway.
6. The Web Services Gateway routes the message to the appropriate trading partner Web service.

DI server flow - inbound

This section describes how business partners engage in process and data sharing using an EDI-VAN paradigm. The Web Services Gateway is not used in this scenario; instead, messages flow into the Business Connection Enterprise Edition via the DI Server.

1. An EDI request arrives at the enterprise.
2. The DI server performs the tasks of:
   a. Enveloping.
   b. Splitting.
   c. Standards-based validation.
   d. Functional acknowledgement.
   e. Conversion into a serialized XML data stream.
3. The MQ connector agent uses event polling to detect an incoming EDI service request.
4. The MQ connector calls the XML data handler, to transform the EDI document into a business object.
5. The instantiation of the business object triggers the execution of a collaboration.
6. The application connector controller invokes a mapping function to transform the generic business object into an application-specific business object.
7. The application connector controller sends the application-specific business object to the application connector agent, which interacts with the application.

DI server flow - outbound

This section continues the example shown in the previous section, demonstrating how the message flows out through the DI server.

1. The response data from the application is mapped back to a business object and sent back to the collaboration via the MQ connector.
2. The collaboration sends the requested data back through the MQ connector to convert the results (through the use of the EDI data handler).
3. The XML data handler transforms the business object into a serialized XML data stream and sends it to the DI Server.
4. The DI Server:
   a. Packages the document by:
      1) Converting the XML data stream into the required EDI format.
      2) Creating appropriate header information, including the sender and receiver identification.
      3) Encoding the document and header into a single message.
   b. Sends the message to the specified trading partner via the VAN.

DI Server flow - other transport infrastructures

In addition to VANs, the DI server is designed to interface with other transport infrastructures via MQSeries(R) queues. Other transports could include the EDI-INT standards AS1 and AS2. In such scenarios, the alternative transport provides or accepts EDI messages from the external trading partner on behalf of the DI server and forwards those messages to or from the DI server. The transport infrastructure is responsible for authentication and authorization. DI functions and flows are otherwise the same as described in the previous section on DI inbound and outbound flows.

An example of an interface with an external transport is integration with the TPI server. See your IBM representative for more information.

Where to next?

If you will be writing Web services, see the Web Services documents, which provide instructions on how to build and deploy a Web service.

If you want to read about the programming interfaces available, see the Using Business Connection APIs document and the related Javadocs.

To obtain information on the Document Transfer collaborations and artifacts (in case, for example, you want to modify them), see the Document Transfer Collaboration document. The Document Transfer collaborations are used by the Document Exchange service.

To obtain information on the Registration Flow collaboration and artifacts (in case, for example, you want to modify them), see the Registration Flow Collaboration document. The Registration Flow collaboration is used by the Registration and Provisioning service.

Notices

IBM may not offer the products, services, or features discussed in this document in all countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

WebSphere Business Connection Lab Director
IBM RTP Laboratory
3039 Cornwallis Road
P.O. BOX 12195

Raleigh, NC 27709-2195
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Programming interface information

Programming interface information is intended to help you create application software using this program.

General-use programming interfaces allow you to write application software that obtain the services of this program's tools.

However, this information may also contain diagnosis, modification, and tuning information. Diagnosis, modification and tuning information is provided to help you debug your application software.

Warning: Do not use this diagnosis, modification, and tuning information as a programming interface because it is subject to change.

Trademarks and service marks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM
alphaWorks
AIX
CrossWorlds
DB2
DB2 OLAP Server
DB2 Universal Database
DeveloperWorks
MQSeries
SecureWay
WebSphere

Lotus is a trademark of International Business Machines Corporation and Lotus Development Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.