

*IBM WebSphere Business Integration Connect
Enterprise Edition and Advanced Edition*



Architecture Overview

Version 4.2.0

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices and Trademarks” on page 19.

First Edition (September 2003)

This edition applies to Version 4, Release 2, Modification 0, of IBM[®] WebSphere[®] Business Integration Connect Advanced Edition (5724-E75) and Enterprise Edition (5724-E87), and to all subsequent releases and modifications until otherwise indicated in new editions.

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

*IBM Burlingame Laboratory
Information Development
577 Airport Blvd., Suite 800
Burlingame, CA 94010
U.S.A*

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 anyway it believes appropriate without incurring any obligation to you.

© Copyright International Business Machines Corporation 2003. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. About this book	5
Chapter 2. Introduction	7
Chapter 3. The architecture	9
Receiver	9
Community Console	9
Document Manager	9
Document Processing Engine	10
State Engine	10
Alert Engine	10
Delivery Manager	10
Archive Manager	11
Database and shared storage	11
Profile	11
Gateways	11
Connections	11
Certificates and IP addresses	12
Alerts configuration	12
Non-Repudiation data repository	12
Message data repository	12
Document activity	12
Document events	13
Summary data	13
Communication via JMS	13
Scaling characteristics	13
Message flows	15
RosettaNet Notification PIP from community participant	15
EDI document from community participant over AS2	16
Integration with WebSphere InterChange Server and WebSphere Data Interchange	18
Chapter 4. Notices and Trademarks	19
Notices	19
Programming interface information	21
Trademarks and service marks	21

Chapter 1. About this book

This document describes the high-level architecture for IBM® WebSphere® Business Integration Connect Version 4.2.0. It describes the components that make up WebSphere Business Integration Connect and provides examples of the flow of business documents throughout the system.

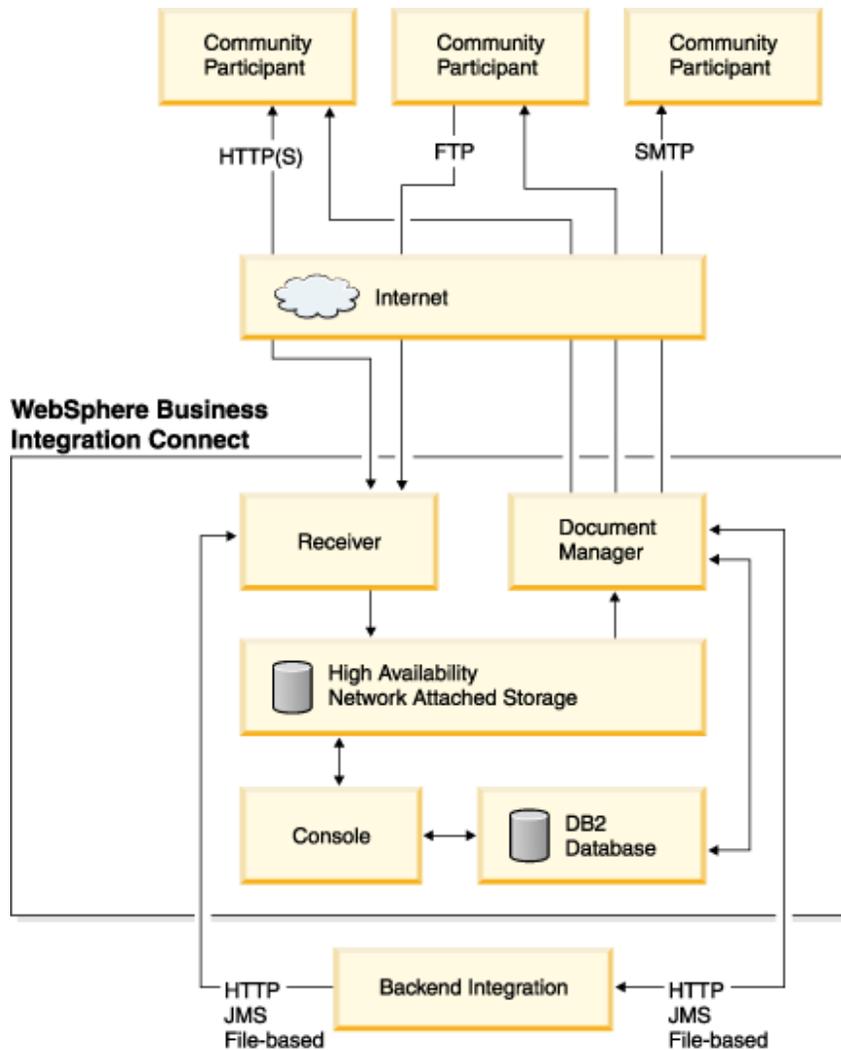
Chapter 2. Introduction

Business Integration Connect enables business-to-business (B2B) process integration and data sharing among partners of all types and sizes. It can be thought of as a suite of distributed, multi-tier Java™ applications that are architected from the ground up to exploit the benefits of Java 2 Enterprise Edition (J2EE) and that work together to provide B2B functionality to enterprises. Each Business Integration Connect offering instance is deployed on a per enterprise/community basis. Enterprises and their partners benefit from reduced integration costs and faster deployment of new processes and services, using open standards and proven technology.

The three major components of Business Integration Connect are the Receiver, the Console, and the Document Manager.

- The Receiver handles secure and reliable receipt of documents, independent of transport protocol.
- The Console is a standard J2EE component that provides Community Console access to the Community Manager and to the Community Participant. The Community Manager can view the entire community. Participants have a more limited view, which allows them to modify their profiles and monitor the flow of their documents.
- The Document Manager function is delivered as one or more routing and processing servers, which implement the reliable routing, validation, and translation services within Business Integration Connect. When validation is required, it is performed through the use of XML schemas. When translation is required, it is performed through the use of both XSLT-based transformations and Java-based transformations, optimized for performance.

The following illustration shows how the components work together:



WebSphere Business Integration Connect components

The components and their relationship to the database and shared storage are described in more detail in the sections that follow. Note that all components must have access to the same file system.

Chapter 3. The architecture

The following sections describe the major components of WebSphere Business Integration Connect.

Receiver

The Receiver component accepts documents from community participants and from backend applications and stores them. Specifically, it:

- Receives a document over a supported transport protocol
- Writes the document and metadata relating to the document to shared file services

The Receiver records any transport-specific data (for example, the source IP address and certificate information about the SSL connection) to the metadata file and completes any transport-specific technical acknowledgment (for example, sending a 200 response to an HTTP POST).

Community Console

The Community Console is the window into the business process activity and error resolution. It primarily has three different users: the Community Operator, the Community Manager, and the Community Participant. The console provides role-based access control to the various features and views. The features of the console include:

- Tools to quickly connect to the community
- Business-process event and exception monitoring
- Detailed reports and analysis on business process, trend, and exception activity
- Tools to troubleshoot document processing
- Ability to drill down to events and raw documents

The Community Console is a J2EE application and is implemented using JSPs, servlets and Enterprise Java Beans (EJBs). Data is persisted and retrieved from the various stores using the EJB interface and also direct JDBC calls.

Document Manager

The Document Manager retrieves stored data, processes it, and routes it, both to Community Participants and to enterprise applications. Specifically, it:

1. Reads the raw document and metadata and saves the inbound document to the non-repudiation directory on shared storage

2. Saves the inbound document location pointer to the database via JDBC
3. Processes the data to the destination format (performing tasks such as validation and transformation, if specified) and saves the destination document to the non-repudiation directory on shared storage
4. Saves the destination document metadata and document location pointer to the database via JDBC
5. Delivers data to its intended destination, which could be a JMS queue, a directory, or a URL

The following sections describe how the subcomponents of the Document Manager perform the tasks presented in the previous list.

Document Processing Engine

The Document Processing Engine performs all of the processing of messages. The Document Processing Engine is responsible for:

- Unpacking messages
- Ensuring that the source of the message is authorized
- Filtering out duplicate messages
- Validating the structure and content of the message
- Translating the message into the format required by the destination
- Packaging the message for the destination, including digitally signing and encrypting the message, if needed
- Storing both the original inbound message and the final outbound message in the non-repudiation repository
- Passing the packaged message to the outbound transport engine

State Engine

The State Engine encapsulates the business rules on a per-protocol basis and executes instructions based on those rules (for example, initiating a retry when no acknowledgment has been received in the defined interval).

Alert Engine

The Alert Engine monitors activity and generates e-mail notifications. You can configure the Alert Engine to indicate which alerts are generated, to whom the alerts are sent, and when the alerts are delivered.

Delivery Manager

The Delivery Manager component is responsible for transporting messages to specific destinations, maintaining a separate queue of messages for each destination. A dedicated transport mechanism exists for each destination, so problems delivering to one destination should not affect transport to other destinations.

Archive Manager

All documents received, and all documents sent by Business Integration Connect, are archived for the purposes of non-repudiation and display to users. The Archive Manager is responsible for moving data from the online storage to permanent media after a certain time period (which you can configure).

Database and shared storage

Business Integration Connect maintains configuration and operational data in a variety of persisted data stores. Generally the data store is in the form of a relational database; however, there are certain items stored in other ways (for example, the raw message data in the non-repudiation and message stores is kept on the file system).

Profile

The participant profile consists of configuration data that is used in document routing and console access. The profile information can be classified as:

- Partner data
- Users
- Contacts

The participant data includes participant names and business identifiers such as DUNS numbers. A single partner can have one or more business identifiers.

Users are accounts that have access to the Business Integration Connect console. A user account gets its permissions to interact with the console based on group membership.

Contacts are similar to users; however, they cannot interact with the console, and they can only be configured to receive notifications from the alert system.

Gateways

Gateways describe the endpoints of a defined connection. When a document is routed through Business Integration Connect, the connection is referenced and the target gateway definition is returned to the system. The gateway defines destination URIs, optional login information, and transport-level settings.

Connections

Connections define valid interactions between Community Participants. They consist of information about the type of document protocol, document type, source participant, target participant, connection type, and source and target gateways. The Document Manager uses the information in the connection to determine if translation is required and to determine the destination gateway information.

Certificates and IP addresses

The following security information is stored:

- Certificate information used to certify the sending Community Participant based on the client certificate used during the SSL connection
- Certificates used for encryption and digital signature validation
- IP addresses used to confirm that the source IP address is correct and that documents can be posted from that address

Alerts configuration

Alerts are defined at a partner level and consist of a variety of attributes to describe event-based alerts or volume alerts.

You can define event-based alerts so that they will be triggered each time the event occurs or so that they will be batched, based on an interval. You can also configure the alert with a contact list for notification based on a defined schedule.

Non-Repudiation data repository

Documents are stored on a disk that has shared access from all components of Business Integration Connect (Receiver, Console, and Document Manager). The original document (as it was received) and the final document (as it was sent) are stored.

Message data repository

Documents are also stored in a console-readable format. This disk also has shared access from all components of Business Integration Connect (Receiver, Console, and Document Manager).

Document activity

Business Integration Connect logs information to describe documents as they are routed. Details are logged about the document as it was received and as it was transmitted. The following types of information are logged:

- The format of the document, including the source participant, target participant, source protocol, target protocol, source document type, and target document type
- Metrics that describe the size of the document and the time it was received and transmitted
- The state that is logged against a document, describing whether it was transmitted to the participant

Document events

Business Integration Connect uses events to track activities and logs the events in a central event log. The events, which are classified as Informational, Warning, or Critical Errors, can be generated by different components in Business Integration Connect.

Events can be tied back to document activity when they are in relation to a document that was routed by Business Integration Connect. The events can also track non-document related activities, such as logging into the system.

Summary data

Business Integration Connect summarizes key metrics, which can be displayed in the console. The information that is summarized includes:

- The number of documents received, sent, and failed. These counts are summarized by certain attributes by hour.
- The number of events that were generated. These counts are summarized by certain detail attributes by hour.

These counts are rolled up by hour and can be correlated back to the document activity logs.

Communication via JMS

Communication between some components is done via JMS. JMS queues with reliable storage allow the flexibility of locating components on different machines while still maintaining a standard inter-component communication method.

Scaling characteristics

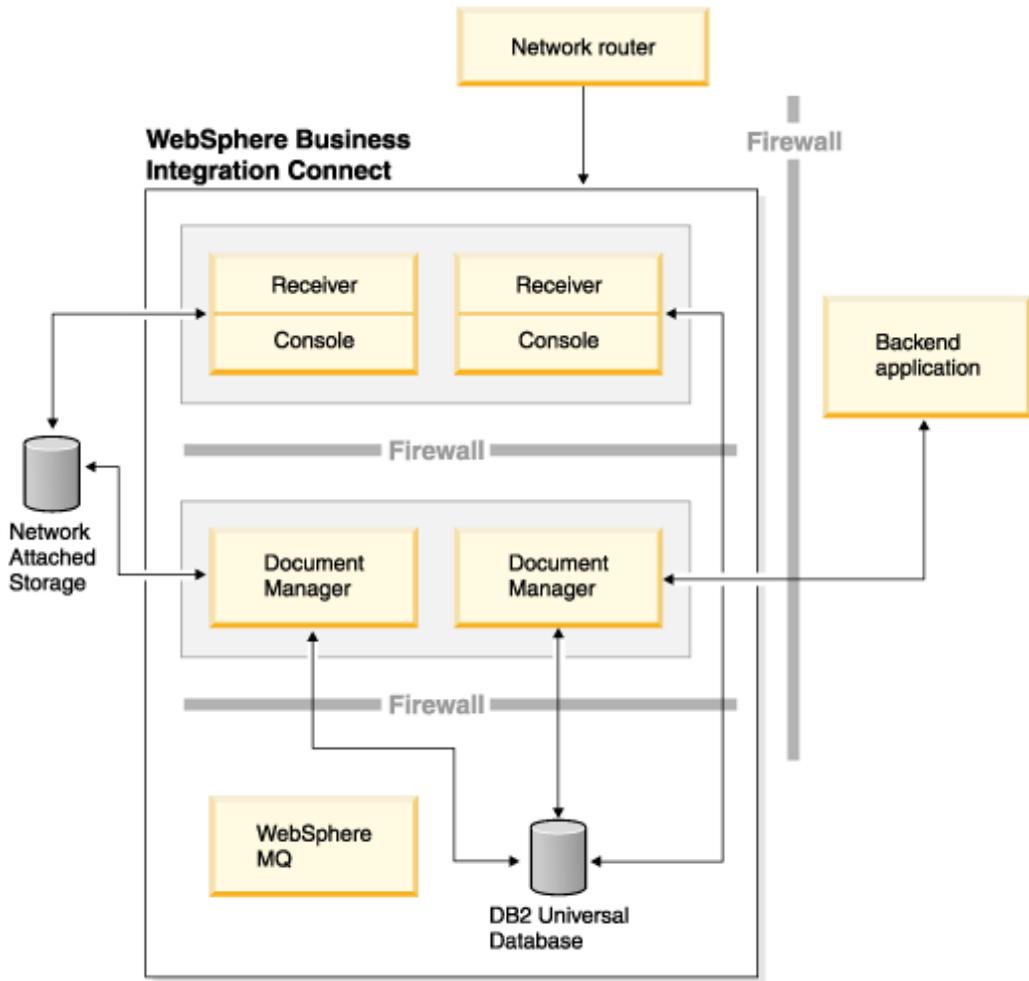
All of the components in Business Integration Connect are designed to scale both horizontally and vertically, where horizontal scaling is characterized as running multiple instances across multiple servers, and vertical scaling is characterized as adding processing power to a single server.

To support vertical scaling, each component is implemented as a multithreaded Managed Bean, and each MBean is deployed within the Business Integration Connect runtime. Each Business Integration Connect runtime is configured with the MBeans that will run in that instance, and the MBeans can be configured for the number of processing threads the MBean creates. Because each Business Integration Connect runtime instance represents a separate Java virtual machine (JVM), having the ability to group MBeans within a single runtime allows tuning the memory configuration of a single JVM to the needs of the MBeans configured to run in that instance. Likewise, MBeans with vastly different JVM configuration needs can be segregated into different JVMs and tuned appropriately.

To support horizontal scaling, each component is architected around the same design principle. Components pick up a unit of work from a shared work queue, thereby locking it so other instances do not pick up the same unit of work. When processing is completed, the results are recorded, the item is removed from the shared input work queue, and the work unit is passed to the next component through another shared work queue mechanism. Work queues are shared among all instances of the component, even if multiple instances of the component are running on different servers. Shared work queues are implemented as persistent JMS queues.

Having the ability to run any of the components on any server allows for greater flexibility with regard to scaling the system. I/O or compute-intensive components can be deployed on servers optimized for their needs. Also, the shared work queue mechanism allows each component to scale independently of other components.

The following illustration shows a typical deployment in which two instances each of the Receiver, Console, and Document Manager components are installed.



Sample Business Integration Connect configuration

Message flows

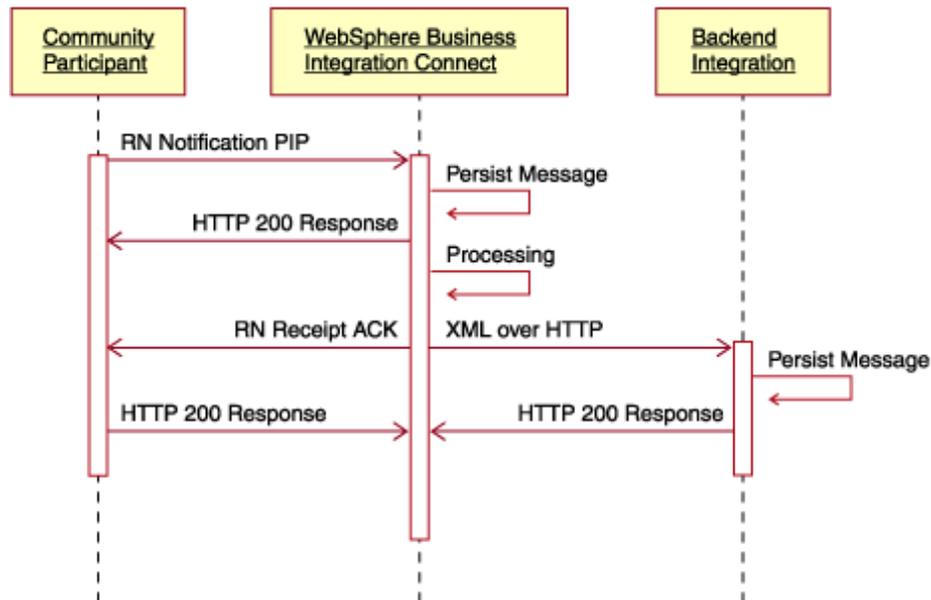
This section describes how the components interact by way of describing the flow of a message through the Business Integration Connect server.

RosettaNet Notification PIP from community participant

This example assumes a community participant sends a RosettaNet Notification PIP to the Business Integration Connect, which performs the following steps:

1. Validates the inbound PIP
2. Translates the PIP to a custom XML format (if required)
3. Delivers the translated XML to a backend application via HTTP
4. Returns a receipt acknowledgment to the community participant

The following illustration shows the high-level sequence diagram.



Sequence diagram of the delivery of a RosettaNet PIP

The following is a description of the operations that happen within WebSphere Business Integration Connect as a result of the sequence shown in the previous illustration:

1. The Receiver component receives the initial HTTP POST from the Community Participant.
2. The Receiver saves the message to disk.
3. The Receiver returns an HTTP 200 response to the initiator. This step is executed because this is an asynchronous exchange of messages between the participants.
4. The Document Manager picks up the message for processing by the Document Processing Engine.

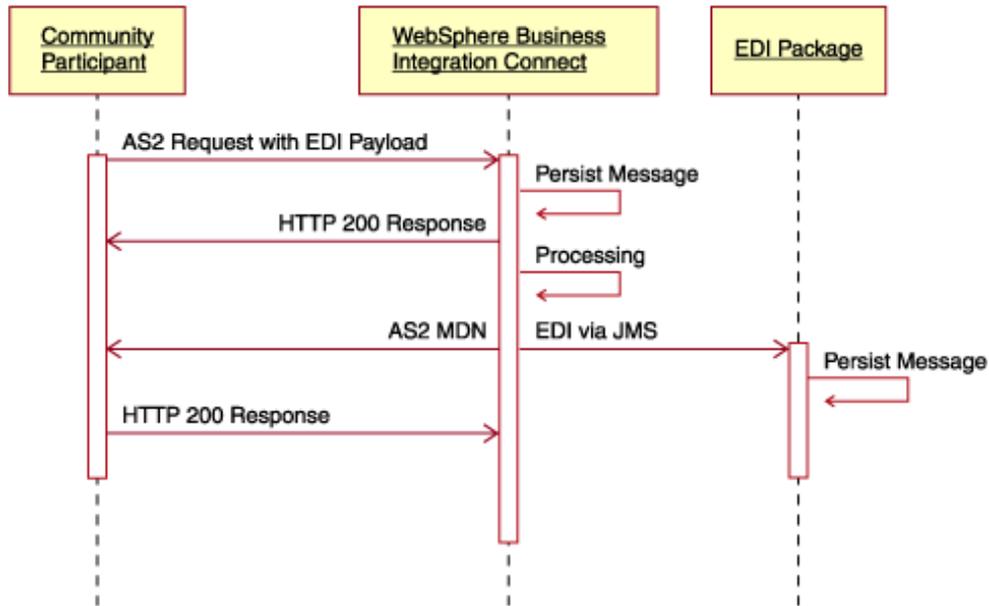
5. The Document Manager saves the original message to the non-repudiation database.
6. The Document Processing Engine unpacks the message (performing decryption, de-enveloping, and structure-level validation).
7. The Document Processing Engine stores the decrypted message.
8. The Document Processing Engine performs authentication and authorization checks.
9. The Document Processing Engine performs a duplicate check.
10. The Document Processing Engine performs content-level validation.
11. The Document Processing Engine records the RosettaNet state.
12. If required, the Document Processing Engine performs translation to the custom XML format.
13. If designated by the configuration, the Document Processing Engine digitally signs or encrypts the message (or both).
14. The Document Processing Engine adds header information that is to be sent to the destination.
15. The Document Processing Engine saves the final packaged message to the non-repudiation database.
16. The Delivery Manager queues the message for delivery.
17. The Delivery Manager checks the destination status.
18. If the destination status is online, the Delivery Manager sends the message.
19. The State Engine updates the business process state to reflect that the message was successfully sent.

After the RosettaNet state is recorded, the following sequence of events is initiated. These steps are shown in a separate list because they happen in parallel with the activity after step 11. Given the asynchronous nature of these two chains of processing, there is no guarantee of the order in which the steps will be executed.

1. The business process state is polled for new work.
2. A new inbound RosettaNet message is found.
3. A receipt acknowledgment payload is created.
4. The receipt acknowledgment payload is passed to the main input queue for packaging and delivery.
5. Business Integration Connect picks up the new message (receipt acknowledgment) and processes it as usual, eventually delivering the receipt acknowledgment to the initiating participant. Essentially this is entering the processing at step 4.

EDI document from community participant over AS2

This sample flow describes the actions that occur when Business Integration Connect processes an EDI file received over the AS2 protocol from a Community Participant. It is assumed that the inbound AS2 message requested an asynchronous MDN. A high-level depiction of this sample is shown in the following sequence diagram:



Sequence diagram of the delivery of an EDI document

1. The Receiver component receives the initial HTTP POST from the Community Participant.
2. The Receiver saves the message to disk.
3. The Receiver returns an HTTP 200 response to the initiator.
4. The Document Manager picks up the message for processing by the Document Processing Engine.
5. The Document Manager saves the original message to the non-repudiation database.
6. The Document Processing Engine unpacks the message (performing decryption, de-enveloping, and structure-level validation).
7. The Document Processing Engine stores the decrypted message.
8. The Document Processing Engine performs authentication and authorization checks.
9. The Document Processing Engine performs a duplicate check.
10. The Document Processing Engine performs content-level validation.
11. The Document Processing Engine records the process state.
12. If designated by the configuration, the Document Processing Engine digitally signs or encrypts the message (or both).
13. The Document Processing Engine adds header information that is to be sent to the destination.
14. The Document Processing Engine saves the final packaged message to the non-repudiation database.
15. The Delivery Manager queues the message for delivery.

16. The Delivery Manager checks the destination status.
17. If the destination status is online, the Delivery Manager sends the message.
18. The State Engine updates the business process state to reflect that the message was successfully sent.

Once step 11 is completed, the following sequence of events is initiated. These steps are shown in a separate list because they happen in parallel with the activity after step 11. Given the asynchronous nature of these two chains of processing, there is no guarantee of the order in which the steps will be executed.

1. The business process state is polled for new work.
2. A new inbound EDI message is found.
3. An MDN payload is created.
4. The payload is passed to the main input queue of the Business Integration Connect server for packaging and delivery.
5. Business Integration Connect picks up the new message (receipt acknowledgment) and processes it as usual, eventually delivering the receipt acknowledgment to the initiating participant. Essentially this is entering the processing at step 4.

Integration with WebSphere InterChange Server and WebSphere Data Interchange

Business Integration Connect typically routes a document from a Community Participant to a backend application, where the processing of the document occurs. Similarly, the backend application might generate the document, in which case Business Integration Connect routes the document to the Community Participant.

Business Integration Connect can also route documents to and receive documents from WebSphere InterChange Server or WebSphere Data Interchange. For example, you might have a collaboration within WebSphere InterChange Server set up to handle purchase orders. Business Integration Connect receives the purchase order from the Community Participant and routes it to the InterChange Server for eventual transport to the collaboration.

The interaction between Business Integration Connect and the InterChange Server can be over the HTTP, HTTPS, or JMS transport protocol.

The interaction between Business Integration Connect and WebSphere Data Interchange is over the JMS transport protocol.

The [Integration Overview](#) describes how Business Integration Connect can be used in conjunction with the InterChange Server and WebSphere Data Interchange.

Notices and Trademarks

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:

IBM Burlingame Laboratory Director
IBM Burlingame Laboratory
577 Airport Blvd., Suite 800
Burlingame, CA 94010
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
the IBM logo
CrossWorlds
DB2
DB2 Universal Database
MQSeries
Passport Advantage
WebSphere

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

Intel and Xeon are trademarks or registered trademarks of Intel Corporation in the United States, other countries, or both.

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

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

