

IBM CrossWorlds
WebSphere Business Integration for
Retail Distribution



Solution Overview

Version 4.1.1

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices and Trademarks” on page 21.

First Edition (October 2002)

This edition applies to Version 4, Release 1, Modification 1, of *IBM® CrossWorlds®* (5724-C12) 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.

© Copyright International Business Machines Corporation 2002. All rights reserved.

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

Contents

Introduction	1
Industry trends and challenges	1
Solution description	1
Architecture	3
Types of integration	3
Solution architecture	4
Components	7
Business objects	7
Collaborations	7
Generic collaborations	8
Solution-specific collaborations	10
Connectors	11

IBM MQSeries Workflow	12
Data mapping templates	12

Use case	13
Actors	13
Course	13

Example task flow	15
------------------------------------	-----------

Notices and Trademarks	21
Notices	21
Programming interface information	22
Trademarks and service marks	23

Introduction

IBM® WebSphere® Business Integration for Retail Distribution portfolio enables retailers and their suppliers to leverage process and application integration. By utilizing WebSphere Business Integration business process solutions, which include templates designed specifically to meet their needs, these companies can meet the challenges of global competition and adapt to increasingly rapid market changes. IBM's WebSphere Business Integration for Retail Distribution solutions can improve operational efficiency, enable faster application implementation, and reduce the costs involved with modern retailing.

Industry trends and challenges

The retail industry is facing many challenges in the competitive business environment of the new century. Evolving market forces, from a sluggish economy to increasingly intense competition, are leading retailers to revisit their business models.

Retail companies must focus on reducing costs and minimizing the time required to bring products to market. The need for integration software is growing quickly as companies require management of high volumes of transactions, while attempting to automate and optimize their businesses at the process level. Integration can enable automation of tedious, error-prone tasks, reducing labor and error-tracking costs. Integration across multiple sales channels can improve the customer experience and increase brand loyalty. However, retailers require process integration that is specific to their industry, and in order to implement this integration, different companies must be able to rely on standardized data.

Before retailers can effectively use product data in internal systems such as store operations, order management, warehouse management, logistics management, and merchandising, both supplier and retailer data must be updated and conform with an industry standard. But trading partners typically do not have a systematic feedback mechanism with each other to know when the product data in their respective computer systems is actually correct and synchronized.

The emergence of UCCnet, a third-party external exchange that provides product registry services enabling synchronized item and location data between trading partners, has brought new opportunities for manufacturers and retailers. UCCnet can facilitate both the exchange of standards-compliant data and end-to-end business process management. IBM's latest addition to the WebSphere Business Integration for Retail Distribution portfolio, the Product Information Management for Retailers solution, enables retailers to synchronize their product data both inside and outside of their enterprise.

Solution description

The Product Information Management for Retailers solution, the newest business process solution to be added to the WebSphere Business Integration for Retail Distribution portfolio, addresses retailers' needs by helping them to:

- Maximize the results from their e-business initiatives as well as from investments in applications and systems.
- Accelerate their product-to-shelf times.

- Decrease costs by reducing or eliminating errors associated with manual re-keying of product data received from their suppliers.

Business integration based on IBM e-business technology not only enables manufacturers, suppliers, and retailers to improve the quality of item data, but opens additional possibilities for enterprise collaboration and process optimization that can generate increased revenue, reduce costs, and provide competitive advantage.

To process inbound data, the solution utilizes modules called *collaborations*, which contain the business logic. Included in the solution are generic, cross-industry collaborations, together with highly customized collaborations created specifically for the solution. Detailed information about these collaborations is included in the section “Collaborations” on page 7.

By utilizing a series of collaborations, together with business objects, connectors, maps, and managed MQSeries® Workflow (which automates retailers’ item data review/approval processes), the following tasks can be successfully automated:

- Sending and receiving data to and from the UCCnet standard registry
- Validating that the data meets the retailers’ standards, helping manage the item approval process
- Storing and retrieving the item data during the item review/approval process
- Publishing approved item data to other enterprise systems

The result is a highly efficient solution that can jump-start integration into the UCCnet registry, reduce operational costs and errors, increase sales by providing the accurate, timely data needed for educated business decisions, and increase market share by accelerating product-to-retail shelf cycles.

Architecture

WebSphere Business Integration for Retail Distribution solutions provide a scalable framework that allows businesses to interconnect multiple disparate processing entities (such as applications, systems, and human resources) within an enterprise context. The solutions connect data and processing entities in a way that fully leverages existing applications and integrates them with new applications. The resulting efficiencies in total system integration provide increased return on investment and business value to the enterprise.

The Product Information Management for Retailers solution is based upon WebSphere technology, which provides pre-built integration components, including collaborations, an integration hub, maps, connectors, and business objects, enabling swift integration of disparate applications. The integration hub is a scalable, reliable, and secure environment for business integration, providing the following benefits:

- A centralized, common view of enterprise-wide data and processes
- A comprehensive set of technical services
- A logical, distributed hub-and-spoke architecture
- Data and messaging format independence

For a detailed description of the integration hub, refer to the following documents:

- IBM CrossWorlds Technical Introduction
- IBM CrossWorlds System Implementation Guide

Types of integration

WebSphere Business Integration facilitates two main types of integration:

- Information Connectivity and Integration
- Process Integration

Information Connectivity and Integration (ICI) is the coupling of heterogeneous application information and data. Connectivity between applications is a fundamental starting point for Information Integration solutions. As connectivity configurations and topologies grow in complexity, additional considerations involve the management of information and data delivery from one application to another (for example, routing rules and data transformation). ICI provides routing and data integration functionality. It routes its information between endpoints, leaving decisions on what actions to take next to the senders and receivers. ICI focuses on *where* data is located and handles data normalization.

Process Integration (PI) is the implementation of internal and external business processes in a way that fully utilizes IT systems to add efficiency and flexibility within an enterprise. Process Integration includes functions required for traditional business process management solutions. Typically, PI solutions begin with a focus on processes internal to the enterprise and evolve to include processes that fully integrate external partners. This external integration requires functions that support business-to-business interaction management. PI focuses on *what* data is needed and *how* that data is used.

Solution architecture

With the Product Information Management for Retailers solution, internal and external collaborative processes are linked throughout the operations of an enterprise. It provides the platform, connectors, and tools to integrate applications, data models, and process workflows, creating a more efficient, uniform, and responsive business environment, both within the enterprise and between the enterprise and its trading partners. The architecture is based on layers of functionality listed below in order from top-most layer to bottom-most layer:

Solution Studio layer

This top-most layer contains tools such as the IBM CrossWorlds® Business Object Designer, Process Designer, Map Designer, Relationship Designer, and System Manager.

Industry Libraries layer

This layer contains three generic, or cross-industry, artifacts including the IBM CrossWorlds Collaboration for UCCnet Message Manager, IBM CrossWorlds Collaboration for Merchandise Qualification, and IBM CrossWorlds Collaboration for Data Storage, together with additional, highly customized artifacts created specifically for this solution.

Runtime Solutions layer

Controlled within this layer are B2B integration, user experience integration, process choreography, application/information integration, and the interaction of logical components in any particular solution deployment (through the integration hub).

Common Solution and Integration Services layer

Controlled within this layer are the directory, security, messaging, audit log, exception, process management, portal, and gateway.

WebSphere Platform layer

This layer holds WebSphere Application Server, MQSeries, the database, and MQSeries Workflow.

Infrastructure layer

This layer's information is hosted in data centers (network, operating system, hardware).

The Solution Studio provides a common build environment for applications and solutions. The artifacts that make up the Product Information Management for Retailers solution within the Industry Libraries layer consist of collaborations, application adapters, message sets, and business objects. The collaborations are built on Runtime Solutions layer components, including the integration hub, in any particular solution deployment. In this architecture, the ICI layer (contained within Common Solution and Integration Services) provides the connection between the business process definitions, located within the Runtime Solutions layer, and the WebSphere platform and existing infrastructure, which hold the business functions operating on legacy systems.

ICI executes the state transitions, required by Process Integration (contained within the Runtime Solutions layer) to traverse the business processes that drive the integration architecture, in a way that allows these business processes to be reused. The primary goal of ICI is to provide loose coupling of the business functions to the business process so that either or both can be independently enhanced over time.

To achieve this goal, ICI must provide message routing and transformation services against a granular set of messages in the particular message model utilized between PI and ICI. This approach ensures that the interactions that leverage the coupling of the functions and the navigation are a normalized implementation of the message vocabulary, and as such, are independent of the systems providing the functions. By taking this approach, systems can be integrated without needing to be continually adjusted as other aspects of the topology are changed. This approach also simplifies the architecture because it positions all process choreography within the PI layer, so that it reflects all aspects of the business process.

Process integration requirements involving only automated processing are addressed using IBM integration technology in the form of layered collaborations, which are discussed in detail in the section “Collaborations” on page 7.

Components

The Product Information Management for Retailers solution is composed of business objects, collaborations, connectors, maps, and managed workflow. These basic components work together to enable a demand-side trading partner to receive item data from a supply-side trading partner through a global data registry such as UCCnet, and to process the data for completeness, approval, and eventual update to back-end systems. The solution also delivers appropriate responses to the registry.

Note: The following sections outline the basic Product Information Management for Retailers solution components at a high-level. For descriptions of hardware and software prerequisites, as well as detailed product component configuration, see the Installation Guide.

Business objects

Business objects are logically equivalent to messages. Each contains a transaction type (i.e., Create, Update) and associated data, and is communicated across various interfaces to multiple processing entities in a solution (i.e., collaborations or managed workflow). Multiple business objects are used to transform data from one format to another and to trigger processing events as the business processes progress. The Product Information Management for Retailers solution includes the following business objects:

- UCCnet_envelope, which holds item data and attributes in the format presented by the UCCnet 2.1 Standard eXtensible Markup Language (XML) Document Type Definition (DTD).
- UCCnetGBO_envelope business object, which is a form of the UCCnet_envelope business object with certain predefined attribute values.
- Retail_Item, which includes item data and attributes in the format required for processing by the individual retailer (including retailer-specific extended item attributes).
- Various business objects used to facilitate temporary data storage during processing.

Collaborations

Automated machine-to-machine processing is performed by *collaborations*. Collaborations manage the flow and the execution of business process and integration logic within the solution. They are bound to connectors or other collaborations and subscribe to specific events triggered by business objects.

In the Product Information Management for Retailers solution, several collaborations are used to exchange data with UCCnet, manage requests for additional supplier information, validate the contents of the data, route the data through a review or approval process, temporarily store data during review/approval processing, and synchronize the validated data with other applications throughout the enterprise. Collaborations run on an integration hub.

The Product Information Management for Retailers solution is made up of three, generic collaborations:

- “IBM CrossWorlds Collaboration for UCCnet Message Manager”
- “IBM CrossWorlds Collaboration for Merchandise Qualification” on page 9
- “IBM CrossWorlds Collaboration for Data Storage” on page 9

Each of these collaborations contains one or more collaboration templates at the application/information integration level.

Several solution-specific collaborations are also supplied:

- “Solution storage collaborations” on page 10, including IdentifierStore, MessageStore, and ItemStore collaborations built from the generic IBM CrossWorlds Collaboration for Data Storage
- “Role_Email collaboration” on page 11

The following sections describe the uses of the collaborations included in the Product Information Management for Retailers solution.

Generic collaborations

The Product Information Management for Retailers solution utilizes generic, or cross-industry, artifacts as well as highly customized artifacts created specifically for this solution. The IBM CrossWorlds Collaboration for UCCnet Message Manager, IBM CrossWorlds Collaboration for Merchandise Qualification, and IBM CrossWorlds Collaboration for Data Storage collaborations are examples of non-solution specific artifacts, which can be easily utilized in solutions other than the Product Information Management for Retailers solution. See “Solution-specific collaborations” on page 10 for details on the custom artifacts of the solution.

IBM CrossWorlds Collaboration for UCCnet Message Manager

This collaboration is composed of a set of two technical collaborations. These collaborations handle all dialog necessary to communicate with UCCnet, providing a streamlined process for receiving XML data from the UCCnet standard product registry, for initiating processing of this data, and for sending responses back to UCCnet. The two technical collaborations include the following:

UCCnetMessageReceive

The UCCnetMessageReceive collaboration handles and filters incoming messages. The incoming XML document is in the form of a UCCnet worklist. This worklist contains multiple messages to be processed. Due to the need to properly respond to each message, as well as to ensure that no data loss occurs, UCCnetMessageReceive parses this worklist into the individual messages and enables persistence of each message in a message store. The Product Information Management for Retailers solution utilizes the IBM CrossWorlds Collaboration for Data Storage to implement this persistence in a message store. The collaboration allows customizations based on individual customer situations to be added before the item data contained in the message is transformed into a Retail_Item business object. It also prevents duplicate items from being processed by enabling persistence of an item identifier to an identifier store. The Product Information Management for Retailers solution utilizes the IBM CrossWorlds Collaboration for Data Storage to implement this persistence in an identifier store. See the document UCCnetMessageReceive Collaboration for detailed information on this collaboration.

UCCnetMessageSend

The UCCnetMessageSend collaboration handles outgoing messages. It accepts a Retail_Item business object and, based on the value of a status

attribute in the business object, responds appropriately to UCCnet. See the document UCCnetMessageSend Collaboration for detailed information on this collaboration.

IBM CrossWorlds Collaboration for Merchandise Qualification

This collaboration is composed of a set of three technical collaborations. These collaborations validate the item data, help manage the item approval process, and route the approved item information to other enterprise systems. The technical collaborations include the following:

ItemValidation Collaboration

The ItemValidation collaboration validates the item data to ensure it is correct, complete, and of the proper format. It accepts or rejects a business object based on customized business policy rules, evaluates an accepted business object based on a customized list of required attribute data, and directs the business object to the appropriate processing port based on the results of the evaluation. If the data does not pass the validation processes, the collaboration can automatically enable notification to an appropriate user, stating that additional information is needed to continue with the approval process. It can also be configured to log the processed business object and to enable persistence of the business object to a local item store. The Product Information Management for Retailers solution utilizes the IBM CrossWorlds Collaboration for Data Storage to implement this persistence in a local item store. See the document ItemValidation Collaboration for detailed information on this collaboration.

ItemCollector Collaboration

The ItemCollector collaboration builds a complete business object by merging partial business objects containing updated information obtained from other collaborations or connectors, such as review/approval information from MQSeries Workflow, with a complete copy of the business object retrieved from a local item store. The Product Information Management for Retailers solution utilizes the IBM CrossWorlds Collaboration for Data Storage to implement this retrieval of the business object from the local item store. The collaboration copies the updated attribute values from the partial objects into the retrieved one through the use of configuration properties. See the document ItemCollector Collaboration for detailed information on this collaboration.

Process_Reviewed_Item Collaboration

The Process_Reviewed_Item collaboration synchronizes a business object to back-end applications based on the status of the object following the review/approval process. The collaboration also sends the business object to a port to notify the source application of the status of the processed business object. It can also be configured to log the processing business object, to enable notification to an appropriate user of the status of business object processing, and to enable saving a business object in or deleting a business object from a persistent item store. The Product Information Management for Retailers solution utilizes the IBM CrossWorlds Collaboration for Data Storage to implement this persistence or deletion of the business object to and from the local item store. See the document Process_Reviewed_Item Collaboration for detailed information on this collaboration.

IBM CrossWorlds Collaboration for Data Storage

This collaboration, known by the technical name of DataStore, manages the processes used to temporarily store, update, retrieve, and delete data during the review and approval processes. As a result, required item information is available

to the trading partner as long as an item is being processed, even if the approval process spans days, weeks, or even months.

This cross-industry collaboration persists the incoming business object in XML format, while allowing for retrieval, updating, and deletion through either auto-generated keys or user-specified keys based on business object attributes. It can be configured to accommodate any business object. See the document *DataStore Collaboration* for detailed information on this collaboration.

Note: The Product Information Management for Retailers solution utilizes this collaboration for storing item identifiers, messages, and the items, themselves. To simplify solution configuration, three collaboration templates built from the DataStore collaboration are included with the Product Information Management for Retailers solution. See the section “Solution storage collaborations” for more information on these templates.

Solution-specific collaborations

The Product Information Management for Retailers solution utilizes highly customized artifacts created specifically for this solution, in addition to generic, or cross-industry, artifacts. The IdentifierStore, MessageStore, and ItemStore solution storage collaborations, as well as the Role_Email collaboration, are examples of solution-specific artifacts. See “Generic collaborations” on page 8 for details on the generic artifacts associated with the Product Information Management for Retailers solution.

Solution storage collaborations

To simplify configuration, three collaboration templates built from the DataStore collaboration are included with the Product Information Management for Retailers solution. All are preconfigured to accept and serialize a specific business object into XML and store it in a database table. Each collaboration also enables the stored business object to be retrieved, updated, or deleted. The output of each is bound to a JDBC connector, which actually performs the storage operations. They are identified by the following technical names:

IdentifierStore collaboration

The IdentifierStore collaboration serializes a UCCnetGBO_identifier business object. In the context of the solution, this collaboration is called by the UCCnetMessageReceive and UCCnetMessageSend collaborations to store, retrieve, update, or delete the item’s unique identifier from the identifier store database. Having the item’s unique identifier stored prevents the solution from processing duplicate items received from UCCnet. See the document *IdentifierStore Collaboration* for detailed information on this collaboration and the document *UCCnetGBO_identifier Business Object* for detailed information on this business object.

MessageStore collaboration

The MessageStore collaboration serializes a UCCnetGBO_storable business object. In the context of the solution, this collaboration is called by the UCCnetMessageReceive and UCCnetMessageSend collaborations to store, retrieve, update, or delete the message header information in the message store database. Having the message header information stored enables the message to be used in return messages. See the document *MessageStore Collaboration* for detailed information on this collaboration and the document *UCCnetGBO_storable Business Object* for detailed information on this business object.

ItemStore collaboration

The ItemStore collaboration serializes a Retail_Item business object. In the context of the solution, this collaboration is called by the ItemValidation and Process_Reviewed_Item collaborations to store the Retail_Item in an item store database, by the ItemCollector collaboration to retrieve the Retail_Item, and by the Process_Reviewed_Item collaboration to delete the Retail_Item. Having the Retail_Item stored enables the business object to be persisted for use by other parts of the solution while long-running review/approval processes complete. See the document ItemStore Collaboration for detailed information on this collaboration and the document Retail_Item Business Object for detailed information on this business object.

Role_Email collaboration

The Role_Email collaboration is used to extract the email message text, subject text, and recipient or list of recipients from configurable attributes in a triggering business object, and use those attribute values as input to the IBM CrossWorlds sendEmail API. The attributes of the business object can contain the actual message text, subject text, or address(es), or point to filenames that contain those values. Use of filenames to hold values allows email message text, subject text, and recipients to be shared among multiple collaborations. For instance, by enclosing the actual attribute value identifying an email address within a filename, an email message can be sent to a Role, which corresponds to the filename. The actual email address contained in the file can be easily changed as ownership of the Role changes, without needing to make any changes to the collaborations requesting the email. See the document Role_Email Collaboration for detailed information on this collaboration.

Connectors

Connectors are the interfaces between collaborations and applications. Connectors monitor applications for events and alert subscribing collaborations. They provide the interfaces over which business objects are communicated and over which application-specific functionality is integrated into the overall solution. The Product Information Management for Retailers solution requires the following connectors:

- IBM CrossWorlds Connector for TPI, which provides connectivity between the Trading Partner Interchange (TPI) EDI Gateway and the other components of the Product Information Management for Retailers solution.
- IBM CrossWorlds Connector for JDBC, which provides connectivity between the collaborations included in the Product Information Management for Retailers solution and the data persisted in the MessageStore, IdentifierStore, and ItemStore database tables during processing.
- IBM CrossWorlds Connector for E-mail, which is used to facilitate the sending of e-mail notifications throughout the business process, such as error notifications and those requesting additional data from suppliers.
- IBM CrossWorlds Connector for MQSeries, which allows IBM CrossWorlds collaborations to asynchronously exchange business objects with applications that issue or receive MQSeries messages. Such an application is MQSeries Workflow (MQWF) when it sends and receives a user-defined program execution server (UPES) message. This message, in MQ Workflow XML message format, contains, among other data, the data as defined in the MQWF container.
- IBM CrossWorlds Connector for MQ Workflow, which facilitates the transfer of data from the integration hub to the MQSeries Workflow review/approval process as required.

- IBM CrossWorlds Connector for JText, which can be used to transfer data to the demand-side trading partner's master catalog, product information management systems, or legacy systems.

IBM MQSeries Workflow

The *IBM MQSeries Workflow (MQWF)* is used to manage the long-running review/approval processing. This Product Information Management for Retailers solution can be enhanced to accommodate human interactions within it by interacting with a user interface engine, such as WebSphere Portal Server. The Product Information Management for Retailers solution requires MQWF process templates for the review/approval process. The MQWF process templates can be used as models for creating a customized process. These must have the capability to change the status of the item to Approved or Rejected (based on business process requirements), and are often implemented with a user interface (such as an implementation of WebSphere Portal Server). Due to the potentially complex nature of the item data, the only item data currently passed to the review/approval process is that necessary for retrieval of the item from the item store. Since this review/approval process can be highly customized, the customized code must have the necessary item information to perform a proper retrieval of the needed item data.

Data mapping templates

Several *data mapping templates* are included in the Product Information Management for Retailers solution, which can be used to transform data between business objects as necessary. See the Installation Guide for a list of maps included with the solution.

Use case

The Product Information Management for Retailers solution enables a demand-side trading partner to receive item data from the UCCnet standard registry, and to process the data for completeness, approval, and eventual update to back-end systems. The solution also delivers appropriate responses to the registry.

Actors

The Product Information Management for Retailers solution use case has multiple actors detailed in the table below.

Table 1. Use case actors

Actor	Description
Retailer	The establishment participating as the demand-side trading partner in the exchange of product information.
Approver	A member or employee of the retail establishment charged with determining which item data is to be forwarded to back-end enterprise systems.

Course

The following table describes the main course that can be followed in the Product Information Management for Retailers solution use case.

Table 2. Use case description (main course)

#	Activity by the actor	System activity
M1	Retailer uses the UCCnet GUI or some customized process to request Item Publications from UCCnet.	UCCnet sends a message containing publication commands. Collaborations parse the message from UCCnet, build a Retail_Item business object, validate the item, and send the item to MQWF for the review/approval process.
M2	Approver uses a user interface (UI) with MQWF to change the status of the item to Approved or Rejected based on the Retailer's business process.	Following the Approver's response, if the item is approved, it is sent to the back-end application and a response is sent to UCCnet. If the item is rejected, it is not sent to the back-end application and a response is sent to UCCnet.

Example task flow

The following steps provide the high-level flow of how the solution handles an PUB_RELEASE_NEW_ITEM flow. For more detailed process logic behind the steps in this high-level flow, refer to the Overall process logic sections of the following documents:

- For steps 1 to 9, UCCnetMessageReceive collaboration.
- For steps 5 and 6, IdentifierStore collaboration.
- For steps 7, 8, 30, 31, 34, and 35, MessageStore collaboration.
- For steps 10 to 20, ItemValidation collaboration.
- For steps 14, 15, 21, 22, 24, 25, 27, and 28, ItemStore collaboration.
- For steps 21 to 23, ItemCollector collaboration.
- For steps 24 to 28, Process_Reviewed_Item collaboration.
- For steps 29 to 38, UCCnetMessageSend collaboration.

Also, refer to the "Creating the collaboration objects and making port connections" section of the Installation Guide for detailed information on collaboration-to-collaboration and collaboration-to-connector port connections.

Note: Throughout this section, the terms NULL and BLANK are defined as true responses when the attributes are tested using the IBM CrossWorlds business object methods isNull() and isBlank(), respectively. The method isNull() returns true when a value has never been set in an attribute. The method isBlank() returns true when the attribute contains a zero-length string. An attribute containing a space character is not considered BLANK by the isBlank() method.

1. A worklist is requested from UCCnet.
2. The TPI connector receives the worklist, which can contain one or more publication requests, from the TPI server. The TPI connector sends the worklist to the IBM CrossWorlds XML Data Handler, which converts it into a UCCnet_envelope business object. This business object contains the entire UCCnet message, including each individual data instance and the commands related to it.
3. Because the UCCnetMessageReceive collaboration subscribes to the UCCnet_envelope business object, the TPI connector sends it to the collaboration. The UCCnetMessageReceive collaboration receives the UCCnet_envelope business object on its FromTPI port. Before it is received, however, the UCCnet_envelope business object is first converted to a UCCnetGBO_envelope business object by passing through the UCCnet_envelope_to_UCCnetGBO_envelope map.
4. The UCCnetMessageReceive collaboration parses the message, separating each instance of an Item Publication command.
5. Because this is an example of an Item Publication with a PUB_RELEASE_NEW_ITEM command, the UCCnetMessageReceive collaboration accommodates persisting the UCCnetGBO_envelope business object to a local identifier store by first converting it to a UCCnetGBO_identifier business object and then passing it with a Create verb to the IdentifierStore collaboration through its ToIdentifier_Store port.
6. The IdentifierStore collaboration receives the business object on its From port and, through a series of interactions with the IBM CrossWorlds XML Data

Handler, stores the identifier in the identifier store database. The key to this record in the identifier store is made up of the three attributes, `gtin`, `version`, and `topic`, of the `UCCnetGBO_identifier` business object. If the item is already found in the identifier store, it is rejected.

7. The `UCCnetMessageReceive` collaboration accommodates persisting the `UCCnetGBO_envelope` business object to a local message store by first converting it to a `UCCnetGBO_storable` business object and then passing it with a `Create` verb to the `MessageStore` collaboration through its `ToMessage_Store` port.
8. The `MessageStore` collaboration receives the business object on its `From` port and, through a series of interactions with the IBM CrossWorlds XML Data Handler, stores the message in the message store database. The key to this record in the message store is saved in the `correlationID` attribute of the `UCCnetGBO_storable` business object.
9. The `UCCnetMessageReceive` collaboration can filter the business object to ensure that it contains required attributes, that it came from an approved set of vendors, or that it belongs to an approved set of item categories. Based on whether the business object passes or fails this analysis, the collaboration directs processing, as follows:
 - If the object fails analysis, the collaboration converts the `UCCnet_envelope` business object into a `Retail_Item` business object containing the individual Item Publication command and its data. It attaches a status to the object preconfigured in its `FILTER_FAIL_RESPONSE` property and sends it to its `ToRetail_Response` port to be passed to the `UCCnetMessageSend` collaboration for transmittal to UCCnet.
 - If the object passes analysis, the collaboration converts the `UCCnet_envelope` business object into a `Retail_Item` business object containing the individual Item Publication command and its data. It attaches a status of `Pending` to the object and sets the command type according to the original message type received from UCCnet. It then sends this object to its `ToRetail_Processing` port to be passed to the `ItemValidation` collaboration.
10. For the purposes of this example, assume this object passed analysis. The `ItemValidation` collaboration accepts the `Retail_Item` business object on its `From` port.
11. The `ItemValidation` collaboration executes any code added to the template to evaluate the `Retail_Item` business object according to customized business policy rules. The business policy logic code must change the item status to either a value of `Pending` or `Rejected`, which directs further processing by the collaboration, as follows:
 - If the `Retail_Item` business object status is not set to `Rejected` by the business policy logic, the `ItemValidation` collaboration continues to process it.
 - If the `Retail_Item` business object status is set to `Rejected` by the business policy logic, the collaboration aborts any further processing and returns the business object to the caller through its `From` port.
12. For the purposes of this example, assume that the `Retail_Item` business object's status was not set to `Rejected` by the business policy logic. The `ItemValidation` collaboration then checks that those particular attributes of the business object that the user has specified must contain data are not `NULL` or `BLANK`. These specified attributes are determined to be missing if they are `NULL` or `BLANK`. Based on the results of this check, the collaboration handles the `Retail_Item` business object, as follows:

- If all required attribute data is present, the ItemValidation collaboration continues to process it.
 - If the business object is missing data for any specified Retail_Item attribute, the collaboration adds the attribute name to another Retail_Item business object list attribute, which is specified in the collaboration's CUST_DATA_MISS_ATTR configuration property (by default, internals.customer_data_missing_attributes) and continues to process the object.
13. The ItemValidation collaboration executes any customized code the user has added to the template for the command or message type associated with the business object.
 14. If configured to do so, the ItemValidation collaboration enables persistence of the business object to a local item store by sending it with a Create verb to its LocalItemStore port to be passed to the ItemStore collaboration.
 15. The ItemStore collaboration receives the business object on its From port and, through a series of interactions with the IBM CrossWorlds XML Data Handler, stores the item in the item store database. The key to this record in the item store is saved in the internals.correlationID attribute of the Retail_Item business object.
 16. The ItemValidation collaboration sets the business object's status to Pending. It processes the object according to whether it is missing data for any specified Retail_Item attribute, as follows:
 - If the object is missing required attribute data, the collaboration passes it to its ToMissingData port to trigger a process for obtaining the missing data. The ToMissingData port can be connected to another collaboration, or to an asynchronous process for collecting the missing attribute values.
 - If all required attribute data is present, the collaboration passes it to its To port to trigger the IBM MQSeries Workflow (MQWF) process for reviewing/approving the item.
 17. At this point, the flow becomes asynchronous. The long-running business review/approval process begins when the ItemValidation collaboration passes the Retail_Item to the MQ Workflow connector over its To port. The business object passes through the Retail_Item_to_MQWF_Retail_Item map. The MQ Workflow connector builds a Workflow container and invokes a specific MQWF process.
 18. The Approver(s) responsible for approving the item set the item status to Approved, Rejected, or Error through a customized user interface with MQWF.
 19. The MQWF process sends the Workflow container, with the updated status, to the MQSeries connector. At this point, the long-running business review/approval process is complete and the flow becomes synchronous again.
 20. The MQSeries connector maps the MQWF container back into a partial Retail_Item business object by passing it through the Retail_MQWF_WfMessage_to_Retail_Item map and invokes the ItemCollector collaboration.
 21. The ItemCollector collaboration accepts the partial Retail_Item business object on its From port and retrieves the complete copy of the Retail_Item that was stored by the ItemValidation collaboration in the item store database before the review/approval process was started. It retrieves the item by sending the business object and a Retrieve verb to the local_store port to be passed to the ItemStore collaboration.

22. The ItemStore collaboration receives the business object on its From port and, through a series of interactions with the IBM CrossWorlds XML Data Handler, retrieves the item from the item store database and returns it to the ItemCollector collaboration.
23. The collaboration merges the new data received from the MQWF process into the Retail_Item business object that was read from the item store. The merged Retail_Item business object is routed to the Process_Reviewed_Item collaboration via the To port.
24. The Process_Reviewed_Item collaboration receives the Retail_Item business object on its From port. If configured to do so, it enables deletion of the item from the item store database by passing the object with a Delete verb to its local_store port to be passed to the ItemStore collaboration.
25. The ItemStore collaboration receives the business object on its From port and, through a series of interactions with the IBM CrossWorlds XML Data Handler, deletes the item from the item store database.
26. The Process_Reviewed_Item collaboration examines the status of the item and directs processing, as follows:
 - a. If the object status is Pending or Re-process, it routes the Retail_Item business object back to the ItemValidation collaboration via the reprocess port. The Process_Reviewed_Item collaboration waits for a response. When the item is returned on the reprocess port from the ItemValidation collaboration, it is processed, as follows:
 - If the status of the returned item is Pending, it indicates that the item was accepted by the ItemValidation collaboration and sent on to the MQWF review/approval process. At this point, the Process_Reviewed_Item processing ends. When the MQWF process triggered by the ItemValidation collaboration completes, the Retail_Item will again be sent to the Process_Reviewed_Item From port.
 - If the status of the returned item is Rejected or Error, the Retail_Item is sent to the respond_to port to be passed to the UCCnetMessageSend collaboration.
 - b. If the object status is Approved, the Retail_Item business object is routed to the JText connector via the Sync port. The JText connector writes the Retail_Item business object to a file system. The object is also sent to the respond_to port to be passed to the UCCnetMessageSend collaboration.
27. Optionally, the Process_Reviewed_Item collaboration rewrites the Retail_Item to the local item store by sending the business object and a Create verb to the local_store port to be passed to the ItemStore collaboration.
28. The ItemStore collaboration receives the business object on its From port and, through a series of interactions with the IBM CrossWorlds XML Data Handler, stores the item in the item store database.
29. The UCCnetMessageSend collaboration receives the Retail_Item business object on its FromRetail port.
30. The UCCnetMessageSend collaboration extracts the Item Authorization command from the Retail_Item business object. It then retrieves the UCCnet_envelope layer information from the local message store database by converting the Retail_Item to a UCCnetGBO_storable business object and passing it with a Retrieve verb to the MessageStore collaboration through its ToMessage_Store port.
31. The MessageStore collaboration receives the business object on its From port and, through a series of interactions with the IBM CrossWorlds XML Data Handler, retrieves the message from the message store database and returns it to the UCCnetMessageSend collaboration.

32. The UCCnetMessageSend collaboration handles the item according to its status value, its original command type, and the values of certain properties. For the purposes of this example, assume that the original command was a PUB_RELEASE_NEW_ITEM and the status is Approved.
33. The UCCnetMessageSend collaboration composes the UCCnetGBO_envelope business object by first converting the Retail_Item to a UCCnetGBO_RI_S business object and then to a UCCnetGBO_envelope business object. The resulting UCCnetGBO_envelope business object contains all of the information needed for the UCCnet message.
34. The UCCnetMessageSend collaboration enables the deletion of the entry from the message store by converting the Retail_Item to a UCCnetGBO_storable business object and passing it with a Delete verb to the MessageStore collaboration through its ToMessage_Store port.
35. The MessageStore collaboration receives the business object on its From port and, through a series of interactions with the IBM CrossWorlds XML Data Handler, deletes the message from the message store database.
36. The UCCnetMessageSend collaboration passes the UCCnetGBO_envelope business object to the TPI connector through the UCCnetGBO_envelope_to_UCCnet_envelope map on the ToTPI_Response port.
37. The TPI connector calls the IBM CrossWorlds XML Data Handler to produce the XML message and passes this message to the TPI server.
38. The UCCnetMessageSend collaboration returns a success notification to UCCnet.

The Process_Reviewed_Item, ItemValidation, and ItemCollector collaborations can be configured to initiate notification processes of item status or error conditions. Each collaboration has a port that can be connected to the Role_Email collaboration.

Notices and Trademarks

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

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 CrossWorlds 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 necessarily 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 may contain 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 may contain 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.

Programming interface information

Programming interface information, if provided, 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 or registered trademarks of International Business Machines Corporation in the United States or other countries, or both:

IBM
the IBM logo
AIX
CrossWorlds
the CrossWorlds logo
DB2
DB2 Universal Database
MQIntegrator
MQSeries
Tivoli
WebSphere

Lotus, Domino, Lotus Notes, and Notes Mail are trademarks of the Lotus Development Corporation 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 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.

IBM CrossWorlds Servers V4.1.1
IBM CrossWorlds Full Toolset V4.1.1
IBM CrossWorlds Connectors V4.1.1
IBM CrossWorlds Collaborations V4.1.1

