IBM CrossWorlds
WebSphere® Business Integration for
Retail Distribution

Item Validation Collaboration

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.

Contents

ItemValidation collaboration

The ItemValidation collaboration is used to accept or reject a Retail_Item business object based on customized business policy rules, to evaluate an accepted business object based on a customized list of required attribute data, and to direct the business object to the appropriate port based on the results of the evaluation. The ItemValidation collaboration can also be configured to do the following:

- Accommodates saving the business object to a persistent store.
- Log the processed business object.
- Initiate notification of a business object rejection or if a processing error is detected.

Required documents

The ItemValidation collaboration is based on CollaborationFoundation and uses its features, ports, and configuration properties. ItemValidation also has features, ports, and configuration properties that are unique to it.

To create and configure an ItemValidation collaboration object, use the following documents:

- This document for the ItemValidation collaboration-specific information.
- Standard Collaboration Processes for information about business processes inherited from CollaborationFoundation.
- Standard Collaboration Properties for information about configuration properties inherited from CollaborationFoundation.
- Collaboration Development Guide for information about CollaborationFoundation, and for general information about creating and configuring collaboration objects.

Collaboration setup

This section includes the following information:

- "Port information"
- "Setting up the collaboration" on page 3

Port information

The following figure illustrates the ItemValidation collaboration's ports, as they are displayed in IBM® CrossWorlds® System Manager (CSM):

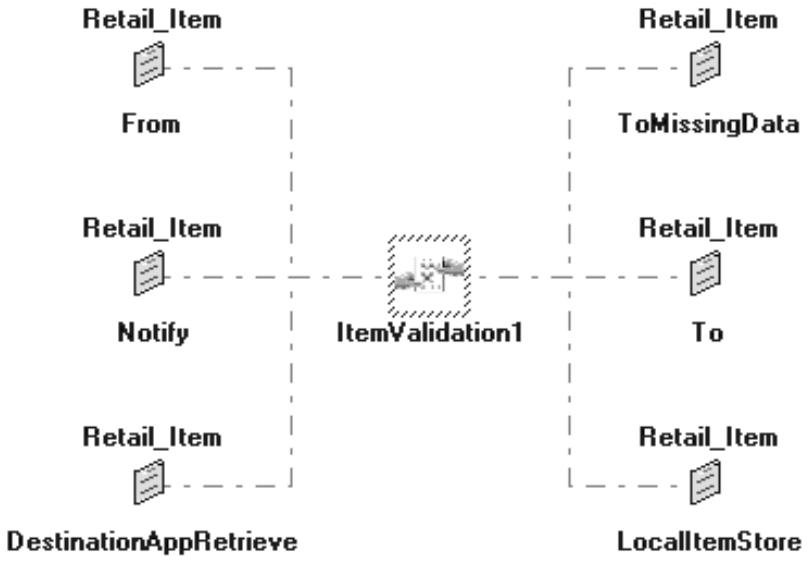

Figure 1. ItemValidation collaboration's ports

Table 1. Port name: From

Business object	Bound to	Function	Verbs used
Retail_Item	Retail_Item data source	Receives the triggering business object. At the end of a synchronous call, this port also returns the triggering business object to the source application with its status attribute indicating the state of the business object processing.	Create

Table 2. Port name: To

Business object	Bound to	Function	Verbs used
Retail_Item	A collaboration or connector that processes the validated item data.	Sends the business object containing all available data to the next processing step.	Create

Table 3. Port name: ToMissingData

Business object	Bound to	Function	Verbs used
Retail_Item	A collaboration or connector that initiates retrieval of missing item data attributes.	Sends the business object containing the names of required but missing attributes to the collaboration or connector that obtains the required attribute data.	Create

Table 4. Port name: Notify

Business object	Bound to	Function	Verbs used
Retail_Item	A collaboration or connector that provides notification processing for rejected items or items in error. This notification step is optional.	Sends the business object containing a message indicating that the object was rejected or an error was detected to a notification process.	Create

Table 5. Port name: LocalItemStore

Business object	Bound to	Function	Verbs used
Retail_Item	A persistent storage mechanism, such as an implementation instance of the IBM CrossWorlds Collaboration for Data Storage.	Sends the business object to be persisted in a storage mechanism.	Create, Delete

Table 6. Port name: DestinationAppRetrieve

Business object	Bound to	Function	Verbs used
Retail_Item	Port connector	This port is currently unused.	Retrieve

Setting up the collaboration

To set up ItemValidation as a stand-alone collaboration object, complete the following steps:

1. Create the ItemValidation collaboration object.
2. Bind the ports as described in "Port information" on page 1.
3. Set the "Configuration properties" on page 11 for ItemValidation.

Synchronization process

This section illustrates the following business processes for this collaboration:

- "Overall process logic" on page 4
- "Inherited process logic" on page 11

Overall process logic

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.

The following flow shows the process logic for this collaboration's Create verb:

1. The ItemValidation collaboration is triggered by the receipt of a Retail_Item business object and a Create verb containing a Create, Update, Delete, or Load message type. This trigger is sent by a message source.

   Note: Message type is equivalent to the value of the Retail_Item business object attribute named in the collaboration configuration property ITEM_COMMAND_ATTRIBUTE (by default, `internals.item_command`). See "ITEM_COMMAND_ATTRIBUTE" on page 15 for more information on this property. Message types include the following:

   Create An item is to be treated as a new submission from the source channel.

   Update An existing item is being updated to reflect changes in data previously submitted.

   Delete An item is no longer available and is to be processed for removal.

   Load The initial loading of items.

2. 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.

   The ItemValidation collaboration is configured to execute business policy logic for any or all Retail_Item business object commands or message types through specific configuration properties. The user must set these properties and add the customized code containing the business policy rules to the ItemValidation collaboration template. See "Validating an object through customized business policy rules" on page 6 for more information on performing these tasks.

   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. The collaboration can initiate notification of the rejection. See "Notifiying of item rejection or processing error conditions" on page 10 for more information on performing this task.

3. If the Retail_Item business object status is not set to Rejected by the business policy logic, the ItemValidation collaboration checks that 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.

This missing data check logic is configured to execute for any or all Retail_Item business object commands or message types through specific configuration properties, which must be set by the user. The user can add customized data check logic to the collaboration template as well. See "Validating an object through requiring data for specific attributes" on page 6 for more information on performing these tasks.

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 CUST_DATA_MISS_ATTR configuration property (by default, `internals.customer_data_missing_attributes`) and continues to process the object. See "CUST_DATA_MISS_ATTR" on page 13 for more information on this property.

4. 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. See "Adding customized code to the collaboration template" on page 9 for more information on performing this task.
5. If configured to do so, the ItemValidation collaboration accommodates persisting the business object to a local store by sending it to its LocalItemStore port. See "Persisting an object to a local item store" on page 9 for more information on performing this task.
6. If the business object is missing data for any specified Retail_Item attribute, the ItemValidation collaboration sets the business object's attribute value named in the collaboration configuration property ITEM_STATUS_ATTRIBUTE (by default, `internals.item_status`) to Pending and passes it to its ToMissingData port to trigger a process for obtaining the missing data. See "ITEM_STATUS_ATTRIBUTE" on page 15 for more information on this property.
7. If all required attribute data is present, the ItemValidation collaboration sets the business object's attribute value named in the collaboration configuration property ITEM_STATUS_ATTRIBUTE to Pending and passes it to its To port to trigger a process for approving the item.

If error conditions are encountered during any stage of processing, the ItemValidation collaboration sets the Retail_Item business object's attribute value named in the collaboration configuration property ITEM_STATUS_ATTRIBUTE to `Error`, logs the error, and returns the object to the calling collaboration through its From port.

The ItemValidation collaboration can log the Retail_Item business object in the configured IBM CrossWorlds InterChange Server (ICS) log destination based on the value of the attribute named in the collaboration configuration property ITEM_STATUS_ATTRIBUTE. Logging is controlled through the configuration properties LOG_ERROR_ITEM, LOG_PENDING_ITEM, and LOG_REJECTED_ITEM. See "LOG_ERROR_ITEM" on page 15, "LOG_PENDING_ITEM" on page 16, and "LOG_REJECTED_ITEM" on page 16 for more information on these properties.

The collaboration can also initiate notification of item rejection after business policy analysis or if any error conditions are detected during processing. See "Notifiying of item rejection or processing error conditions" on page 10 for more information.

Validating an object through customized business policy rules

The ItemValidation collaboration contains a subdiagram called Business Policy Processing into which a user can insert customized code to accept or reject an item based on business policies (such as only accepting items from specific trading partners or in specific item categories). This logic is enabled for specific Retail_Item commands or message types by the values specified in the BUSINESS_POLICY_CMDS configuration property (see "BUSINESS_POLICY_CMDS" on page 12 for more information on this property).

The pre-existing subdiagram logic examines the value for the Retail_Item business object's attribute named in the ITEM_COMMAND_ATTRIBUTE configuration property against the values specified in the BUSINESS_POLICY_CMDS configuration property. If the value for the attribute in the ITEM_COMMAND_ATTRIBUTE property is specified in the BUSINESS_POLICY_CMDS property, the ItemValidation collaboration executes the customized code.

The rules in the customized code must specify whether the Retail_Item is accepted or rejected for subsequent processing by changing the value of its attribute named in the configuration property ITEM_STATUS_ATTRIBUTE. If the object is accepted, the code must change the value to Pending; if the object is rejected, it must change the value to Rejected. The pre-existing subdiagram logic uses this attribute value to determine whether to return the item to the caller or to continue processing it. If the item is rejected, the business policy subdiagram raises an exception to abort the item processing and returns it to the main collaboration scenario, where the item rejection processing is handled.

Validating an object through requiring data for specific attributes

A standard message from the input data source might not contain all of the data required by this user's implementation. In this case, the user can identify certain Retail_Item business object attributes for which data must exist (the attributes must not be NULL or BLANK). To do this, a user must perform the following tasks:

- Specify whether missing data checking logic executes for specific Retail_Item commands or message types by setting values in the REQUIRED_ATTRIBUTE_CMDS configuration property. See "REQUIRED_ATTRIBUTE_CMDS" on page 17 for more information on this property.
- Identify in a text file the fully qualified attribute names requiring data. Each line of the file must contain one required attribute name followed by a line return. See "Examples of required attribute file contents" on page 7 for more information.
- Identify the fully qualified name of the file containing the attribute names requiring data in the REQUIRED_ATTRIBUTE_FILE configuration property. See "REQUIRED_ATTRIBUTE_FILE" on page 17 for more information on this property.

Pre-existing required data check logic examines the value of the Retail_Item business object's attribute, named in the configuration property ITEM_COMMAND_ATTRIBUTE, against the values specified in the REQUIRED_ATTRIBUTE_CMDS configuration property. If the value of the

attribute named in the ITEM_COMMAND_ATTRIBUTE property is specified in the REQUIRED_ATTRIBUTE_CMDS property, the ItemValidation collaboration executes the code.

The collaboration contains a subdiagram called File Missing Attribute Logic that reads a list of fully qualified required attribute names from the file named in the REQUIRED_ATTRIBUTE_FILE property. It employs an external Java™ class called RetailUtility in the Java package `com.ibm.wbi.retail.utils` (see "Using the RetailUtility external Java class" on page 8 for more information) to determine if the required attributes listed in the file contain data in the Retail_Item business object being processed. It then adds the names of any required attributes missing data to a Retail_Item business object attribute named in the configuration property CUST_DATA_MISS_ATTR. This attribute consists of a multiple cardinality array of Retail_Missing_Attributes business objects. If there are any names in this missing attribute child business object array, the ItemValidation collaboration passes the Retail_Item business object to the ToMissingData port to start a process for obtaining the missing data.

Note: The Retail_Item business object must contain an instance of its customer_data child business object for the data check code to successfully check the customer_data child business object attributes.

The ItemValidation collaboration contains a subdiagram called Custom Missing Attribute Logic into which the user can insert customized code. This subdiagram contains examples of the following methods to enhance the algorithm for identifying attributes of the Retail_Item business object that are required but missing data.

Note: These examples do not execute as long as the TEST configuration property is set to `false`. See "TEST" on page 19 for more information on this property. These examples should not be executed as part of production code.

- Add fully qualified attribute names to the Required Attribute Vector, which is processed later in the algorithm.
- Add fully qualified attribute names to the Missing Attribute Vector, which is processed later in the algorithm.
- Modify the name of the file that is normally set from the REQUIRED_ATTRIBUTE_FILE collaboration property.
- Create and add instances of Retail_Missing_Attributes business objects directly to the attribute named in the CUST_DATA_MISS_ATTR configuration property (by default, `internals.customer_data_missing_attributes`).

The Custom Missing Attribute Logic can be executed instead of, or in addition to, the File Missing Attribute Logic.

Examples of required attribute file contents: The required attributes file contains a list of fully qualified attribute names. Each line in the file contains one attribute name. Each child in a hierarchical business object structure described by the attribute name is separated by periods. Any multiple-cardinality child business objects (BusObjArray) in the structure are represented in the element by appending front- and end-bracket symbols ([]) to the name of the child attribute. Single-cardinality child business objects must not contain brackets ([]) in the attribute name. The brackets ([]) characters are required to tell the code that parses through the child business objects in the attribute whether the object is a single business object (BusObj) or a business object array (BusObjArray).

Some example file entries corresponding to Retail_Item business object attributes are as follows:

```
customer_data.vendorAddress
item.itemLinks.childItem[].globalTradeItemNumber.gtin
```

Notice that since the childItem business object is a multiple-cardinality business object array, the [] characters are appended to its name in the fully qualified attribute.

Note: The function of the missing attribute code is to create a list of attributes that exist in the Retail_Item business object being processed, but whose values are NULL or BLANK. If an attribute is part of a hierarchy of parent and child business objects, and any of the attribute's parent business objects does not exist in the Retail_Item, then the missing attribute code does *not* include the attribute in the missing attribute list.

The following examples show how the missing attribute list is populated:

- The attribute customer_data.vendorAddress is included in the required attribute file. The Retail_Item business object that triggers the ItemValidation collaboration does not contain an instance of the customer_data child business object attribute. Therefore, the missing attribute code does not include customer_data.vendorAddress in the missing attribute list.
- The attribute customer_data.vendorAddress is included in the required attribute file. The Retail_Item business object that triggers the ItemValidation collaboration does contain an instance of the customer_data child business object attribute. The vendorAddress attribute is NULL. Therefore, the missing attribute code includes customer_data.vendorAddress in the missing attribute list.
- The attribute item.itemLinks.childItem[].globalTradeItemNumber.gtin is included in the required attribute file. The Retail_Item business object that triggers the ItemValidation collaboration contains two instances of the childItem business object attribute, each of which contains an instance of the globalTradeItemNumber child business object. In each instance, the gtin attribute is NULL. Therefore, the missing attribute code adds the following entries to the missing attribute list:

  ```
  item.itemLinks.childItem[0].globalTradeItemNumber.gtin
  item.itemLinks.childItem[1].globalTradeItemNumber.gtin
  ```

  The missing attribute code includes specific references to all required attributes that exist, but are NULL or BLANK.
- The attribute item.itemLinks.childItem[].globalTradeItemNumber.gtin is included in the required attribute file. The Retail_Item business object that triggers the ItemValidation collaboration contains no instances of the childItem business object attribute. Therefore, the missing attribute code includes no references to the item.itemLinks.childItem[].globalTradeItemNumber.gtin attribute in the missing attribute list.

Using the RetailUtility external Java class: The missing attribute data check functionality of the ItemValidation collaboration uses an external Java class called RetailUtility, located in the Java package `com.ibm.wbi.retail.utils`, to check that the required attributes contain data. It recursively parses each fully qualified attribute string to determine if the value of the attribute is NULL or BLANK.

The class contains two methods:

```
public boolean checkRequiredAttribute(BusObj busObj, Vector attrs, Vector err)  \
throws Exception
public boolean checkRequiredAttribute(BusObj busObj, String attr, Vector err)   \
throws Exception
```

The first method takes as input the Retail_Item business object and two Vectors. The first Vector contains the list of fully qualified attribute names to be checked for NULL or BLANK values. On return, the second Vector contains the list of missing attributes (those containing NULL or BLANK values).

The method returns true if no attribute data is missing; it returns false if any required attribute data is missing. The second method operates the same as the first, except it takes as input a single fully qualified attribute name.

At runtime, if the collaboration's missing attribute data check logic is enabled (the value of the Retail_Item business object's attribute named in the configuration property ITEM_COMMAND_ATTRIBUTE exists in the REQUIRED_ATTRIBUTE_CMDS property), and the required attribute Vector contains elements, this class must be added to a directory or to a jar file contained in one of the following:

- On Windows® 2000 systems, the ICS CLASSPATH, which is set up during the IBM CrossWorlds `start_server.bat` file prior to starting the ICS.
- On AIX® and Solaris systems, the CWCLASSES path, which is set up during the IBM CrossWorlds `CWSharedEnv.sh` file prior to starting the ICS.

Access to this external Java class via the ICS CLASSPATH or CWCLASSES path is also required in order to successfully compile the ItemValidation collaboration template.

Adding customized code to the collaboration template

The ItemValidation collaboration contains a subdiagram called Message Type Processing that can be customized to contain any desired logic based on the Retail_Item message type (that is, the contents of the attribute named in the ITEM_COMMAND_ATTRIBUTE configuration property). This subdiagram is executed if the value of the Retail_Item attribute named in the ITEM_COMMAND_ATTRIBUTE property is contained in the MESSAGE_TYPE_PROCESSING_CMDS configuration property. This subdiagram is executed prior to the item being saved in the local item store and sent to a port for further processing.

Persisting an object to a local item store

The ItemValidation collaboration accommodates the ability to persist the Retail_Item business object to a local store. This ability is enabled by the setting for the RETAIN_ITEM_IN_LOCAL_STORE configuration property (see "RETAIN_ITEM_IN_LOCAL_STORE" on page 18 for more information on this property). The collaboration accommodates persisting the business object by sending it to its LocalItemStore port. The LocalItemStore port can then be bound to a persistence mechanism, such as an instance of the IBM CrossWorlds Collaboration for Data Storage.

If the item is rejected by the business policy code (see "Validating an object through customized business policy rules" on page 6 for more information) or if an error is detected during other ItemValidation collaboration processing, the ItemValidation collaboration does not direct processing to the LocalItemStore port. Instead, it changes the value of the Retail_Item business object's attribute named in the ITEM_STATUS_ATTRIBUTE configuration property to `Error` and returns the item to the calling collaboration.

Notifiying of item rejection or processing error conditions

The ItemValidation collaboration can be configured to enable notification if an item is rejected by business policy rules or if errors are detected during ItemValidation collaboration processing.

The following collaboration properties enable notification if an item is rejected by business policy rules:

- SEND_MAIL_ON_REJECTION (see "SEND_MAIL_ON_REJECTION" on page 19 for more information)
- REJECT_EMAIL_ROLE (see "REJECT_EMAIL_ROLE" on page 17 for more information)
- REJECT_EMAIL_MSG (see "REJECT_EMAIL_MSG" on page 16 for more information)
- REJECT_EMAIL_SUBJECT (see "REJECT_EMAIL_SUBJECT" on page 17 for more information)

The SEND_MAIL_ON_REJECTION and REJECT_EMAIL_ROLE must be configured by the user. If the Retail_Item is rejected and the SEND_MAIL_ON_REJECTION and REJECT_EMAIL_ROLE collaboration properties exist and are not BLANK, the collaboration stores the values of the REJECT_EMAIL_ROLE, REJECT_EMAIL_MSG, and REJECT_EMAIL_SUBJECT in attributes of the Retail_Item business object. The Retail_Item attributes into which these values are stored are identified by the EMAIL_ROLE_ATTRIBUTE, EMAIL_MSG_ATTRIBUTE, and EMAIL_SUBJECT_ATTRIBUTE configuration properties, respectively. (See "EMAIL_ROLE_ATTRIBUTE" on page 14, "EMAIL_MSG_ATTRIBUTE" on page 14, and "EMAIL_SUBJECT_ATTRIBUTE" on page 14 for more information on these properties.) Then the collaboration sends the object to its Notify port. If the REJECT_EMAIL_ROLE property exists but is BLANK, no notification is executed (that is, no default mail recipient exists).

The following collaboration properties enable notification if errors are detected during ItemValidation collaboration processing:

- SEND_MAIL_ON_ERROR (see "SEND_MAIL_ON_ERROR" on page 18 for more information)
- ERROR_EMAIL_ROLE (see "ERROR_EMAIL_ROLE" on page 14 for more information)
- ERROR_EMAIL_MSG (see "ERROR_EMAIL_MSG" on page 14 for more information)
- ERROR_EMAIL_SUBJECT (see "ERROR_EMAIL_SUBJECT" on page 15 for more information)

The SEND_MAIL_ON_ERROR and ERROR_EMAIL_ROLE must be configured by the user. If an error is detected and the SEND_MAIL_ON_ERROR and ERROR_EMAIL_ROLE collaboration properties exist and are not BLANK, the collaboration stores the values of the ERROR_EMAIL_ROLE, ERROR_EMAIL_MSG, and ERROR_EMAIL_SUBJECT in attributes of the Retail_Item business object. The Retail_Item attributes into which these values are stored are identified by the EMAIL_ROLE_ATTRIBUTE, EMAIL_MSG_ATTRIBUTE, and EMAIL_SUBJECT_ATTRIBUTE configuration properties, respectively. Then the collaboration sends the object to its Notify port. If the ERROR_EMAIL_ROLE property exists but is BLANK, no notification is executed (that is, no default mail recipient exists).

If the REJECT_EMAIL_MSG, REJECT_EMAIL_SUBJECT, ERROR_EMAIL_MSG, or ERROR_EMAIL_SUBJECT properties exist but are BLANK, default values are placed in the Retail_Item attributes before the business object is sent to the Notify port.

Inherited process logic

This collaboration inherits the following business processes from the CollaborationFoundation template:

- Retrieve process
- USE_RETRIEVE process
- Filtering data
- Additional Retrieve process
- Email process for error handling

For information on these processes, see Standard Collaboration Processes.

Configuration properties

This section describes the following properties for this collaboration:

- "Standard properties"
- "Collaboration-specific properties"

Standard properties

This collaboration inherits the following standard configuration properties from the CollaborationFoundation template:

- 1_EXCLUDE_VALUES
- 1_FAIL_ON_INVALID_VALUE
- 1_FILTER_ATTRIBUTE
- 1_INCLUDE_VALUES
- ADDITIONAL_RETRIEVE
- CONVERT_CREATE
- CONVERT_UPDATE
- INFORMATIONAL_EXCEPTIONS
- SEND_EMAIL
- USE_RETRIEVE

For information on these configuration properties, see Standard Collaboration Properties.

Note: Do not enable inherited configuration properties within the context of this collaboration. Changing default values for these inherited properties can cause processing to malfunction.

Collaboration-specific properties

This collaboration has the following collaboration-specific configuration properties:

- "BUSINESS_POLICY_CMDS" on page 12
- "COLLAB_PROPERTY_FILE_ATTR" on page 13
- "COLLAB_PROPERTY_FILE_EXT" on page 13
- "COLLAB_PROPERTY_FILE_PATH" on page 13

Note: The properties COLLAB_PROPERTY_FILE_ATTR, COLLAB_PROPERTY_FILE_EXT, COLLAB_PROPERTY_FILE_PATH, and TEST must be used only for development testing purposes. Do not use these properties in production code. Also, 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.

BUSINESS_POLICY_CMDS

This property specifies the values of the Retail_Item business object attribute, named in the ITEM_COMMAND_ATTRIBUTE collaboration property, for which business policy logic is enabled.

Table 7. BUSINESS_POLICY_CMDS configuration property

Possible values	Usage
Create (default) All A comma-separated list of values, which can include Create, Update, Delete, Load, or any value that can be contained in the ITEM_COMMAND_ATTRIBUTE property.	The collaboration performs the business policy logic to accept or reject the Retail_Item business object if the attribute of the Retail_Item named in the ITEM_COMMAND_ATTRIBUTE collaboration property contains any of the values listed in the BUSINESS_POLICY_CMDS property. A value of `All` for the BUSINESS_POLICY_CMDS property causes the collaboration to perform the logic for all values of the attribute named in ITEM_COMMAND_ATTRIBUTE.

COLLAB_PROPERTY_FILE_ATTR

This property specifies the attribute of the Retail_Item business object whose contents are used to form the name of a file to be read by the collaboration containing values to modify the collaboration's property variables. This property is used only if the collaboration property TEST is set to the value `True`. There is no default value for this property.

Note: Use this property only for development testing purposes; do not use it in production code.

COLLAB_PROPERTY_FILE_EXT

This property specifies the file extension to be added to the contents of the attribute named in the COLLAB_PROPERTY_FILE_ATTR property to form the name of a file to be read by the collaboration containing values to modify the collaboration's property variables. This property is used only if the collaboration property TEST is set to the value `True`.

Note: Use this property only for development testing purposes; do not use it in production code.

Table 8. COLLAB_PROPERTY_FILE_EXT configuration property

Possible values	Usage
properties	The default collaboration property file extension.

COLLAB_PROPERTY_FILE_PATH

This property specifies the path to be added to the contents of the attribute named in the COLLAB_PROPERTY_FILE_ATTR property to form the name of a file to be read by the collaboration containing values to modify the collaboration's property variables. This property is only used if the collaboration property TEST is set to the value `True`. There is no default value for this property.

Note: Use this property only for development testing purposes; do not use it in production code.

CUST_DATA_MISS_ATTR

This property specifies the attribute of the Retail_Item business object that contains the array of business objects that identify the attributes contained in the Retail_Item business object being processed, which have been specified as required,

but that contain no data.

Table 9. CUST_DATA_MISS_ATTR configuration property

Possible values	Usage
internals.customer_data_missing_attributes	The default Retail_Item attribute name.

EMAIL_MSG_ATTRIBUTE

This property specifies the attribute of the Retail_Item business object into which the text of a notification message is stored prior to the Retail_Item business object's being sent to the collaboration's Notify port.

Table 10. EMAIL_MSG_ATTRIBUTE configuration property

Possible values	Usage
internals.message_text	The default Retail_Item attribute name.

EMAIL_ROLE_ATTRIBUTE

This property specifies the attribute of the Retail_Item business object into which the email address of the email recipient of a message is stored, prior to the Retail_Item business object's being sent to the collaboration's Notify port.

Table 11. EMAIL_ROLE_ATTRIBUTE configuration property

Possible values	Usage
internals.message_recipient_role	The default Retail_Item attribute name.

EMAIL_SUBJECT_ATTRIBUTE

This property specifies the attribute of the Retail_Item business object into which the text of a message subject is stored, prior to the Retail_Item business object's being sent to the collaboration's Notify port.

Table 12. EMAIL_SUBJECT_ATTRIBUTE configuration property

Possible values	Usage
internals.message_subject	The default Retail_Item attribute name.

ERROR_EMAIL_MSG

This property specifies the value that is placed in the EMAIL_MSG_ATTRIBUTE of the Retail_Item business object when an error is detected during processing. This property can contain the actual message text.

Table 13. ERROR_EMAIL_MSG configuration property

Possible values	Usage
The message text for the notification that an error was detected.	This property holds information that is passed to the Notify port. The use of this information is dependent on that which is connected to that port.

ERROR_EMAIL_ROLE

This property specifies the value that is placed in the EMAIL_ROLE_ATTRIBUTE of the Retail_Item business object when an error is detected during processing. This property can contain the actual email address to which to send the error email message. There is no default value for this property.

ERROR_EMAIL_SUBJECT

This property specifies the value that is placed in the EMAIL_SUBJECT_ATTRIBUTE of the Retail_Item business object when an error is detected during processing. This property can contain the actual subject text.

Table 14. ERROR_EMAIL_SUBJECT configuration property

Possible values	Usage
ItemValidation Error	The default error subject text.

ITEM_COMMAND_ATTRIBUTE

This property specifies the attribute of the Retail_Item business object that contains the command or message type of the business object.

Table 15. ITEM_COMMAND_ATTRIBUTE configuration property

Possible values	Usage
internals.item_command	The default Retail_Item attribute name.

ITEM_IDENTIFICATION_ATTRIBUTE

This property specifies the attribute of the Retail_Item business object that contains a unique identification value for each business object. The value is used to identify the item in logged messages.

Note: The value in the Possible values column includes a space to allow it to fit into the table cell. The actual value does not include a space.

Table 16. ITEM_IDENTIFICATION_ATTRIBUTE configuration property

Possible values	Usage
item.itemInformation.globalTradeItem Number.gtin	The default Retail_Item attribute name.

ITEM_STATUS_ATTRIBUTE

This property specifies the attribute of the Retail_Item business object that contains the status of the business object being processed.

Table 17. ITEM_STATUS_ATTRIBUTE configuration property

Possible values	Usage
internals.item_status	The default Retail_Item attribute name.

LOG_ERROR_ITEM

This property specifies whether the collaboration logs the business object being processed when an error is detected during processing. Note that the detected error is always logged.

Table 18. LOG_ERROR_ITEM configuration property

Possible values	Usage
False (default)	The collaboration does not log the business object in addition to the error log.
True	The collaboration logs the business object in addition to the error log.

LOG_PENDING_ITEM

This property specifies whether the collaboration logs the business object being processed when the object is successfully processed. The Pending item status means that the item continues to be processed for required missing attribute data or approval.

Table 19. LOG_PENDING_ITEM configuration property

Possible values	Usage
True (default)	The collaboration logs the business object, which is successfully processed.
False	The collaboration does not log the business object, which is successfully processed.

LOG_REJECTED_ITEM

This property specifies whether the collaboration logs the business object being processed when the object is rejected by the business policy logic.

Table 20. LOG_REJECTED_ITEM configuration property

Possible values	Usage
False (default)	The collaboration does not log the rejected business object.
True	The collaboration logs the rejected business object.

MESSAGE_TYPE_PROCESSING_CMDS

This property specifies the values of the Retail_Item business object attribute named in the ITEM_COMMAND_ATTRIBUTE collaboration property for which message-type processing is enabled.

Table 21. MESSAGE_TYPE_PROCESSING_CMDS configuration property

Possible values	Usage
Create (default) All A comma-separated list of values, which can include Create, Update, Delete, Load, or any value that can be contained in the attribute named in the ITEM_COMMAND_ATTRIBUTE property.	The collaboration performs the message-type processing code if the attribute of the Retail_Item named in the ITEM_COMMAND_ATTRIBUTE collaboration property contains any of the values listed in the property MESSAGE_TYPE_PROCESSING_CMDS. A value of `All` for the MESSAGE_TYPE_PROCESSING_CMDS property causes the collaboration to perform the logic for all values of the attribute named in ITEM_COMMAND_ATTRIBUTE.

REJECT_EMAIL_MSG

This property specifies the value that is placed in the EMAIL_MSG_ATTRIBUTE of the Retail_Item business object when the item is rejected during business policy processing. This property can contain the actual message text.

Table 22. REJECT_EMAIL_MSG configuration property

Possible values	Usage
The message text for the notification that an item was rejected by business processing.	This property holds information that is passed to the Notify port. The use of this information is dependent on that which is connected to that port.

REJECT_EMAIL_ROLE

This property specifies the value that is placed in the EMAIL_ROLE_ATTRIBUTE of the Retail_Item business object when the item is rejected during business policy processing. This property can contain the actual email address to which to send the rejected email message. There is no default value for this property.

REJECT_EMAIL_SUBJECT

This property specifies the value that is placed in the EMAIL_SUBJECT_ATTRIBUTE of the Retail_Item business object when the item is rejected during business policy processing. This property can contain the actual subject text.

Table 23. REJECT_EMAIL_SUBJECT configuration property

Possible values	Usage
Retail Item Rejected by ItemValidation	The default rejected subject text.

REQUIRED_ATTRIBUTE_CMDS

This property specifies the values of the Retail_Item business object attribute named in the ITEM_COMMAND_ATTRIBUTE collaboration property for which the required attribute missing data check logic is enabled.

Table 24. REQUIRED_ATTRIBUTE_CMDS configuration property

Possible values	Usage
Create (default) All A comma-separated list of values, which can include Create, Update, Delete, Load, or any value that can be contained in the attribute named in the ITEM_COMMAND_ATTRIBUTE property.	The collaboration performs the required attribute missing data check logic if the attribute of the Retail_Item named in the ITEM_COMMAND_ATTRIBUTE property contains any of the values listed in the REQUIRED_ATTRIBUTE_CMDS property. A value of `All` for the REQUIRED_ATTRIBUTE_CMDS property causes the collaboration to perform the logic for all values of the attribute named in ITEM_COMMAND_ATTRIBUTE.

REQUIRED_ATTRIBUTE_FILE

This property specifies the path and name of a file that contains a list of Retail_Item attributes, which if present in the Retail_Item business object, must contain data. That is, these attributes must not be NULL or BLANK. The collaboration checks each of the attributes named in this file and, if any are found to be missing data, adds the missing attribute name to a list contained in the Retail_Item business object. The business object is sent to the collaboration's ToMissingData port. This property has no default value.

RETAIL_MISS_ATTR_NAME

This property specifies the attribute of the Retail_Missing_Attributes child business object that contains the name of any Retail_Item required attributes found to be

present and missing data. As the collaboration executes the required attribute missing data check logic, it creates instances of Retail_Missing_Attributes business objects containing the names of missing Retail_Item attributes and adds these business objects to an array contained in the Retail_Item business object.

Table 25. RETAIL_MISS_ATTR_NAME configuration property

Possible values	Usage
attribute_name	The default Retail_Missing_Attributes attribute name.

RETAIL_MISS_ATTR_TYPE

This property specifies the type of the child business object to be instantiated by the collaboration for each of the Retail_Item required attributes found to be present and missing data. As the collaboration executes the required attribute missing data check logic, it creates instances of this type of business object containing the names of missing Retail_Item attributes and adds these business objects to an array contained in the Retail_Item business object.

Table 26. RETAIL_MISS_ATTR_TYPE configuration property

Possible values	Usage
Retail_Missing_Attributes	The default type of the missing attribute child business object.

RETAIN_ITEM_IN_LOCAL_STORE

This property specifies whether the collaboration directs processing to the LocalItemStore port when the item is successfully processed. The collaboration does not direct processing to the LocalItemStore port if the item is rejected by the business policy logic or if an error is detected during the processing of the item.

Table 27. RETAIN_ITEM_IN_LOCAL_STORE configuration property

Possible values	Usage
True (default)	The collaboration sends the successfully processed Retail_Item business object to its LocalItemStore port.
False	The collaboration does not send the successfully processed Retail_Item business object to its LocalItemStore port.

SEND_MAIL_ON_ERROR

This property specifies whether the collaboration directs processing to the Notify port if an error is detected during the processing of the Retail_Item business object.

Table 28. SEND_MAIL_ON_ERROR configuration property

Possible values	Usage
False (default)	The collaboration does not send the Retail_Item business object to its Notify port if an error is detected during the processing of the item.

Table 28. SEND_MAIL_ON_ERROR configuration property (continued)

Possible values	Usage
True	The collaboration sends the Retail_Item business object containing an error message to its Notify port if an error is detected during the processing of the item and if the ERROR_EMAIL_ROLE property contains a value.

SEND_MAIL_ON_REJECTION

This property specifies whether the collaboration directs processing to the Notify port if the Retail_Item business object is rejected by the business policy logic.

Table 29. SEND_MAIL_ON_REJECTION configuration property

Possible values	Usage
False (default)	The collaboration does not send the Retail_Item business object to its Notify port if the item is rejected by the business policy logic.
True	The collaboration sends the Retail_Item business object containing an item rejected message to its Notify port if the item is rejected by the business policy logic and if the REJECT_EMAIL_ROLE property contains a value.

TEST

This property enables and disables development testing logic built into the collaboration.

Note: Use this property only for development testing purposes; do not use it in production code.

Table 30. TEST configuration property

Possible values	Usage
False (default)	Disable the development testing logic built into the collaboration.
True	Enable the development testing logic built into the collaboration. Do not use this setting for production operation.

UTILITY_CLASS

This property specifies the class name required by the collaboration to perform the required attribute data check logic. The collaboration uses this property to check for the presence of the class before attempting to execute the required attribute data check logic. If the required attribute data check is enabled and this class is not available, the collaboration returns the Retail_Item business object with its status set to `Error`.

Table 31. UTILITY_CLASS configuration property

Possible values	Usage
com.ibm.wbi.retail.utils.RetailUtility	The default external Java class used by the collaboration during the required attribute data check logic.

Viewing collaboration messages

To view an explanation of this collaboration's messages, invoke Message Browser and open the collaboration's message file.

To invoke Message Browser and open the collaboration message file, complete the following actions:

1. In the Start menu, click **Programs > CrossWorlds > Server and Tools > Message Browser**.
2. On the **File** menu, click **Open**.
3. Use the **Look In** field to change the current folder to:
   IBM_CrossWorlds_root_dir\collaborations\messages\ItemValidation.txt

See also

- DataStore Collaboration
- ItemCollector Collaboration
- Process_Reviewed_Item Collaboration
- Retail_Item Business Object

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