

Retek® 10 Integration Bus



Integration Guide

**Retek Customer Order Management 10.0
Message Family Manager Designs**



The software described in this documentation is furnished under a license agreement and may be used only in accordance with the terms of the agreement.

No part of this documentation may be reproduced or transmitted in any form or by any means without the express written permission of Retek Inc., Retek on the Mall, 950 Nicollet Mall, Minneapolis, MN 55403.

Information in this documentation is subject to change without notice.

Retek provides product documentation in a read-only-format to ensure content integrity. Retek Customer Support cannot support documentation that has been changed without Retek authorization.

Corporate Headquarters:

Retek Inc.
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403
888.61.RETEK (toll free US)
+1 612 587 5000

Retek® Integration Bus™ and Retek Customer Order Management™ are trademarks of Retek Inc.

Retek and the Retek logo are registered trademarks of Retek Inc.

©2002 Retek Inc. All rights reserved.

European Headquarters:

Retek
110 Wigmore Street
London
W1U 3RW
United Kingdom

All other product names mentioned are trademarks or registered trademarks of their respective owners and should be treated as such.

Printed in the United States of America.

Switchboard:
+44 (0)20 7563 4600

Sales Enquiries:
+44 (0)20 7563 46 46
Fax: +44 (0)20 7563 46 10



Customer Support

Customer Support hours:

8AM to 5PM Central Standard Time (GMT-6), Monday through Friday, excluding Retek company holidays (in 2002: Jan. 1, May 27, July 4, July 5, Sept. 2, Nov. 28, Nov. 29, and Dec. 25).

Customer Support emergency hours:

24 hours a day, 7 days a week.

Contact Method	Contact Information
Phone	US & Canada: 1-800-61-RETEK (1-800-617-3835) World: +1 612-587-5000
Fax	(+1) 612-587-5100
E-mail	support@retек.com
Internet	www.retek.com/support Retek's secure client Web site to update and view issues
Mail	Retek Customer Support Retek on the Mall 950 Nicollet Mall Minneapolis, MN 55403

When contacting Customer Support:

- Always fill out an Issue Report Form before submitting issues to Retek (request forms from Customer Support if necessary).
- Provide a completely updated Customer Profile.
- Have a single resource per product responsible for coordination and screening of Retek issues.
- Respond to our requests for additional information in a timely manner.
- Use Retek Online Customer Support (ROCS) to submit and update your issues.
- Have a test system in place running base Retek code.

Contents

Chapter 1 – Carrier service (ship method) MFM.....	1
Overview	1
Implementation.....	1
Add to queue procedure.....	1
Retrieve from queue procedure	1
Message family queue	2
Message family sequence	2
Process flow	3
Add to queue.....	3
Get next (message family conforms to the synchronous pattern).....	3
Chapter 2 – Customer order.....	5
Overview	5
Implementation.....	5
Add to queue procedure.....	5
Retrieve from queue procedure	5
Message family queue	6
Message family sequence	6
Process flow	7
addtoq	7
getnxt	7
Chapter 3 – Pending returns	9
Overview	9
Implementation.....	9
Add to queue procedure.....	9
Retrieve from queue procedure	9
Message family queue	10
Message family sequence	11
Process flow	11
Add to queue.....	11
Get next (message family conforms to the synchronous pattern).....	11

Chapter 4 – Reservation backorder..... 13

API summary.....	13
Detail of procedures	13
Add to queue procedure.....	13
Retrieve from queue procedure	13
Message family queue	14
Message family sequence	14
Process flow	14
Addtoq	14
Getnxt	15

Chapter 5 – Customer return..... 17

Overview	17
Package: RCOMMFM_CORETURN	17
Implementation.....	17
Add to queue procedure.....	17
Retrieve from queue procedure	17
Message family queue	18
Message family sequence	18
Process flow	19
addtoq	19
getnxt	19

Chapter 6 – Customer sale 21

Overview	21
Package: RCOMMFM_COSALE	21
Implementation.....	21
Add to queue procedure.....	21
Retrieve from queue procedure	21
Message family queue	22
Message family sequence	22
Process flow	23
addtoq	23
getnxt	23

Chapter 1 – Carrier service (ship method) MFM

Overview

The ‘shipMeth’ message family queue is one of the componets in the shipMeth publish process. It holds all the valid records regarding the publish messages. The information include message sequence number, xml message, publish status and some keys.

The message family manager is a package, which has two functions: add valid ‘shipmeth’ message to queue table and retrieve information from the queue table.

Implementation

A message family manager has two key interfaces, one for queuing messages and one for retrieving messages from the queue.

Add to queue procedure

ADDTOQ (O_status_code, O_error_msg, I_message_type, I_message, I_rdm_dest_id)

The add-to-queue interface is called by event capture trigger EC_table_carrrsv_aiudr, and takes the message type, and, since Ship Method is a synchronously captured messages, the message itself. It inserts a row into the message family queue along with the passed in values rdm_dest_id and the next sequence number from the message family sequence, setting the status to ‘U’npublished. It returns error codes and strings according to the standards of the application in which it is being implemented.

Retrieve from queue procedure

GETNXT (O_status_code, O_error_msg, O_message_type, O_message, O_ship_request_id)

The retrieve interface is publicly exposed procedure and typically called by a RIB publication adaptor. The parameters are well defined and arranged in a specific order.

The message type is the RIB-defined “ShipMethCre”, “ShipMethMod”, and “ShipMethDel”. The message is the xml message, and the rdm_dest_id is the key value for the message. Status code is one of three values, as shown in the following table. For more discussion of the status codes, refer to the Error Handling Guidelines document or the process flow in the following section. These codes come from an EAI team defined RIB_CODES package.

Code	Meaning
E 'MFM_FATAL_ERROR'	A fatal error occurred in the procedure
S 'MFM_SUCCESS'	A message was retrieved
N 'MFM_NO_MSG'	No messages to retrieve at this time

The error text parameter contains application-generated information, such as the application's sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Message family queue

RIB_CARRSRV_MFQUEUE

The message family queue is a table with the following structure:

Column Name	Data Type/Length	Required	Constraints, Indices, Primary Keys	Description
SEQ_NO	NUMBER(15)	YES	PK, I1	sequence number of message in family
PUB_STATUS	VARCHAR2(1)	YES	(U)	publication statuses: unpublished
MESSAGE_TYPE	VARCHAR2(15)	YES		message type
RDM_DEST_ID	NUMBER(10)	YES		key value
MESSAGE	CLOB	YES		xml message

Message family sequence

rib_carrsrv_mfquence_s

Each message family has its own sequence generator that generates sequence numbers for the population of the message family queue's seq_no column.

Process flow

The two interfaces have different process flows, and the retrieve interface follows the Synchronous message pattern.

Add to queue

- Add to queue gets a new sequence number when called and inserts that along with the passed in values into the appropriate table columns, and it sets the published status column to “U” for unpublished.

Get next (message family conforms to the synchronous pattern)

- 1 Retrieve gets the row of data corresponding to the lowest sequence number on the table that is in a “U” status and removes the row from the table, returning the messages and other values via the procedure’s output parameters, with a status code of ‘S’.
- 2 If such record could not be found, the status code will be ‘N’.
- 3 If the procedure encounters an error it updates the status of the message to ‘E’ and returns the MFM_FATAL_ERROR status code. Error information, such as the application’s message sequence number and possible Oracle errors should be returned in the error text.

Chapter 2 – Customer order

Overview

The message family manager for the RCOM customer order publication writes messages from the RCOM system to the queue table (addtoq function), and it also retrieves messages from the queue table to the RIB (getnxt function).

Implementation

A message family manager has two key interfaces, one for queuing messages and one for retrieving messages from the queue.

Add to queue procedure

ADDTOQ (I_message_type, I_ship_request_id, I_physical_wh, I_message)

The add-to-queue interface is called by COE\$PICK_RELEASE, and takes the message type, ship_request_id and, since Customer Order is a synchronously captured messages, the message itself. It inserts a row into the message family queue along with the passed in values and the next sequence number from the message family sequence, setting the status to 'U'npublished. It returns error codes and strings according to the standards of the application in which it is being implemented.

Retrieve from queue procedure

GETNXT (O_status_code, O_error_message, O_message_type, O_ship_request_id, O_physical_wh, O_message)

The retrieve interface is publicly exposed procedure and typically called by a RIB publication adaptor. It's parameters are well defined and arranged in a specific order. The message type is the RIB-defined "COCre", the message is the xml message, and the ship_request_id is the key value for the message. Status code is one of two values, as shown in the following table. For more discussion of the status codes, refer to the Error Handling Guidelines document or the process flow in the following section. These codes come from an EAI team defined RIB_CODES package.

Code	Meaning
MFM_FATAL_ERROR	A fatal error occurred in the procedure
MFM_SUCCESS	A message was retrieved
MFM_NO_MSG	There was no retrievable message on the queue table.

The error text parameter contains application-generated information, such as the application's sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed. The facility id is included in all messages coming from RDM.

Message family queue

RIB_CUSTORDER_MFQUEUE

The message family queue is a table with the following structure:

Column Name	Data Type/Length	Required	Constraints, Indices, Primary Keys	Description
SEQ_NO	NUMBER(15)	YES	PK, I1	sequence number of message in family
PUB_STATUS	VARCHAR2(1)	YES	(U, W, E, S), I1	publication statuses: unpublished, warning, error, skip
MESSAGE_TYPE	VARCHAR2(15)	YES		Message type
SHIP_REQUEST_ID	NUMBER(15)	YES		key value
PHYSICAL_WH	NUMBER(10)	YES		key value
MESSAGE	CLOB	YES		xml message

Notes on the table:

- For queues for families that follow the synchronous pattern, the “W” and “S” allowed value of the publication status check constraint is unnecessary.
- For queues for families that follow the asynchronous pattern, the “S” allowed value of the publication status check constraint is unnecessary, and the message column is not required to be on the table.

Message family sequence

custorder_mfsequence.nextval

Each message family has its own sequence that generates sequence numbers for the population of the message family queue's sequence column.

Process flow

addtoq

- Addtoq inserts a new message into the table rib_custorder_mfqueue. It gets a new sequence number from the rib_custorder_mfqueue_s sequence. The message_type is “COCRe” and pub_status is set to “U” for unpublished.

getnxt

- 1 Retrieve a row from the RIB_CUSTORDE_MFQUEUE corresponding to the lowest sequence number on the table that is in a “U”.
- 2 Removes the selected row from the table, returning the message, message_type, and ship_request_id via the procedure’s output parameters, with a status code of MFM_SUCCESS.
- 3 If the procedure encounters an error it updates the status of the message to E and returns the MFM_FATAL_ERROR status code. Error information, such as the application’s message sequence number and possible Oracle errors should be returned in the error text.

Chapter 3 – Pending returns

Overview

The ‘PendReturn’ message family queue is one of the components in the pend return publish process. It holds all the valid records regarding the publish messages. The information include message sequence number, xml message, publish status and some keys.

The message family manager is a package, which has two functions: add valid ‘pendreturn’ message to queue table and retrieve information from the queue table.

Implementation

A message family manager has two key interfaces, one for queuing messages and one for retrieving messages from the queue.

Add to queue procedure

ADDTOQ (I_message_type, I_return_ord_nbr, I_rma_number, I_item_number, I_physical_wh, I_message)

The add-to-queue interface is called by event capture package ‘rib\$pendreturn_ec’, and takes the message type, and, since Pend return is a synchronously captured messages, the message itself. It inserts a row into the message family queue along with the passed in key values return_order_number, rma_number, item_number, physical_wh and the next sequence number from the message family sequence, setting the status to ‘U’npublished. It returns error codes and strings according to the standards of the application in which it is being implemented.

Retrieve from queue procedure

GETNXT (O_status_code, O_error_msg, O_message_type, O_message, O_return_ord_nbr, O_rma_number, O_physical_wh, O_item_number)

The retrieve interface is publicly exposed procedure and typically called by a RIB publication adaptor. The parameters are well defined and arranged in a specific order.

The message type is the RIB-defined “PendRetCre”, “PendRetDtlMod”, “PendRetDtlCre”, “PendRetDtlDel”, and “PendRetDel”. The message is the xml message, and the return_order_number, rma_number, item_number, physical_wh are the key values for the message. The item_number can be null if the message type is “PendRetDel”. Status code is one of three values, as shown in the following table. For more discussion of the status codes, refer to the Error Handling Guidelines document or the process flow in the following section. These codes come from an EAI team defined RIB_CODES package.

Code	Meaning
E 'MFM_FATAL_ERROR'	A fatal error occurred in the procedure
S 'MFM_SUCCESS'	A message was retrieved
N 'MFM_NO_MSG'	No messages to retrieve at this time

The error text parameter contains application-generated information, such as the application's sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Message family queue

RIB_PendRetrun_MFQUEUE

The message family queue is a table with the following structure:

Column Name	Data Type/Length	Required	Constraints, Indices, Primary Keys	Description
SEQ_NO	NUMBER(15)	YES	PK, I1	sequence number of message in family
PUB_STATUS	VARCHAR2(1)	YES	(U)	publication statuses: unpublished
MESSAGE_TYPE	VARCHAR2(15)	YES		message type
RETURN_ORDER_NUMBER	VARCHAR2(20)	YES		Key value
RMA_NUMBER	VARCHAR2(20)	YES		Key value
ITEM_NUMBER	NUMBER(15)	NO		Key value
PHYSICAL_WH	NUMBER(10)	YES		key value
MESSAGE	CLOB	YES		xml message

Message family sequence

rib_pendreturn_mfquence_s

Each message family has its own sequence generator that generates sequence numbers for the population of the message family queue's seq_no column.

Process flow

The two interfaces have different process flows, and the retrieve interface follows the Synchronous message pattern.

Add to queue

- Add to queue gets a new sequence number when called and inserts that along with the passed in values into the appropriate table columns, and it sets the published status column to "U" for unpublished.

Get next (message family conforms to the synchronous pattern)

- 1 Retrieve gets the row of data corresponding to the lowest sequence number on the table that is in a "U" status and removes the row from the table, returning the messages and other values via the procedure's output parameters, with a status code of 'S'
- 2 If such record could not be found, the status code will be 'N'.
- 3 If the procedure encounters an error it updates the status of the message to 'E' and returns the MFM_FATAL_ERROR status code. Error information, such as the application's message sequence number and possible Oracle errors should be returned in the error text.

Chapter 4 – Reservation backorder

API summary

The message family manager for the RCOM reservation and backorder publication writes messages from the RCOM system to the queue table (addtoq function). It also retrieves messages from the queue table to the RIB (getnxt function).

Detail of procedures

Add to queue procedure

ADDTOQ (I_message_type, I_warehouse, I_item, I_message)

The addtoq interface is called by one of four event capture triggers, and takes the message type, warehouse number, item number and the message itself. It inserts a row into the message family queue along with the passed in values and the next sequence number from the message family sequence rib_coresbo_mfqueue_s. It inserts the status as 'U'npublished. It returns error codes and strings according to the RCOM standards.

Retrieve from queue procedure

GETNXT (O_status_code, O_error_msg, O_message_type, O_message, O_warehouse, O_item)

The retrieve interface is publicly exposed procedure and is called by a RIB publication adaptor. The parameters are well defined and arranged in a specific order according the Message Catalog. There are six different message types: CustBOCre, CustBOCanCre, CustResToBOCre, CustBOtoResCre, CustResCre, and CustResCanCre. The message is the xml message, and the key values for the message are O_warehouse and O_item. Status code is one of two values, as shown in the following table. For more discussion of the status codes, refer to the Error Handling Guidelines document or the process flow in the following section. These codes come from an EAI team defined RIB_CODES package.

Status Code	Description
MFM_FATAL_ERROR	A fatal error occurred in the procedure
MFM_SUCCESS	A message was retrieved
MFM_NO_MSG	No messages to retrieve at this time

The error text parameter contains application-generated information, such as the application's sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Message family queue

RIB_CORESBO_MFQUEUE

The message family queue is a table with the following structure:

Column Name	Data Type/Length	Required	Constraints, Indices, Primary Keys	Description
SEQ_NO	NUMBER(15)	YES	PK, I1	sequence number of message in family
PUB_STATUS	VARCHAR2(1)	YES	(U, W, E, S), I1	publication statuses: unpublished, warning, error, skip
MESSAGE_TYPE	VARCHAR2(15)	YES		message type
WAREHOUSE	VARCHAR2(12)	YES		Key value
ITEM	VARCHAR2(25)	YES		Key value
MESSAGE	CLOB	YES		xml message

Message family sequence

rib_coresbo_mfqueue_s.nextval

There is a unique sequence that is used to order the messages on the queue table.
The sequence starts with 0 and increments up to 9,999,999,999.

Process flow

Addtoq

- 1 The addtoq function simply inserts a new record into the queue table.
- 2 The error handling is the standard error handling for RCOM.

Getnxt

- 1 A cursor retrieves the row of data corresponding to the lowest sequence number on the table that is in a “U” status and removes the row from the table, returning the message and key values via the procedure’s output parameters.
- 2 If the procedure succeeds, it returns a status code `RIB_CODES.MFM_SUCCESS`.
- 3 If the procedure encounters an error it updates the status of the message to “E” and returns the `MFM_FATAL_ERROR` status code. Error information, such as the application’s message sequence number and possible Oracle errors should be returned in the error text.
- 4 If there is no message on the queue table, then the package returns `RIB_CODES.MFM_NO_MSG`.

Chapter 5 – Customer return

Overview

Package: RCOMMFM_CORETURN

The message family manager for the RCOM customer return sale publication writes messages from RCOM to the queue table and retrieves messages from the queue table to the RIB.

Implementation

Add to queue procedure

ADDTOQ (I_message_type, I_store, I_item, I_message)

The add to queue interface is called by the RCOMMFM_CORETURN package, and takes in the message type, store, item and the message itself. It inserts a row into the message family queue including the passed in values and the next sequence number from the message family sequence, setting the status to 'U'npublished.

Retrieve from queue procedure

GETNXT (O_status_code, O_error_message, O_message_type, O_message, O_store, O_item)

The retrieve interface is a publicly exposed procedure and typically called by a RIB publication adaptor. Its parameters are well defined and arranged in a specific order. The message type is the RIB-defined "CustRetSaleCre", the message is the xml message, and the store and item are the key values for the message. Status code is one of four values, as shown in the following table. For more discussion of the status codes, refer to the Error Handling Guidelines document or the process flow in the following section. These codes come from an EAI team defined RIB_CODES package.

Code	Meaning
MFM_FATAL_ERROR	A fatal error occurred in the procedure.
MFM_SUCCESS	A message was retrieved.
MFM_NO_MSG	No message was retrieved.
MFM_WARNING	The next message on the queue cannot be published because a related family member with a lower sequence number is in error status.

The error text parameter contains application-generated information, such as the application's sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Message family queue

RIB_CORETURN_MFQUEUE

The message family queue is a table with the following structure:

Column Name	Data Type/Length	Required	Constraints, Indices, Primary Keys	Description
SEQ_NO	NUMBER(15)	YES	PK, I1	sequence number of message in family
PUB_STATUS	VARCHAR2(1)	YES	(U, W, E, S), I1	publication statuses: unpublished, warning, error, skip
MESSAGE_TYPE	VARCHAR2(15)	YES		message type
MESSAGE	CLOB	YES		xml message
STORE	VARCHAR2(10)	YES		key value
ITEM	VARCHAR2(25)	YES		key value

Message family sequence

RIB_CORETURN_MFQUEUE_S

Each message family has its own sequence that generates sequence numbers for the population of the message family queue's seq_no column.

Process flow

addtoq

- Addtoq inserts a new message into the table rib_coreturn_mfqueue. It gets a new sequence number from the rib_coreturn_mfqueue_s sequence. The message_type is “CustRetSaleCre” and pub_status is set to “U” for unpublished.

getnxt

- 1 Retrieve a row from the RIB_CORETURN_MFQUEUE corresponding to the lowest sequence number on the table that is in a “U”.
- 2 Removes the selected row from the table, returning the message, message_type, store, and item via the procedure’s output parameters, with a status code of MFM_SUCCESS.
- 3 If the procedure encounters an error it updates the status of the message to E and returns the MFM_FATAL_ERROR status code. Error information, such as the application’s message sequence number and possible Oracle errors should be returned in the error text.
- 4 If the next message on the queue cannot be published because a related family member with a lower sequence number is in error status, a status code of MFM_WARNING is returned.
- 5 If there are no messages to retrieve, a MFM_NO_MSG status code is returned.

Chapter 6 – Customer sale

Overview

Package: RCOMMFM_COSALE

The message family manager for the RCOM customer sale publication writes messages from RCOM to the queue table and retrieves messages from the queue table to the RIB.

Implementation

Add to queue procedure

ADDTOQ (I_message_type, I_wh, I_store, I_item, I_message)

The add to queue interface is called by the RCOMMFM_COSALE package, and takes in the message type, warehouse_id, store, item and the message itself. It inserts a row into the message family queue including the passed in values and the next sequence number from the message family sequence, setting the status to 'U'npublished.

Retrieve from queue procedure

GETNXT (O_status_code, O_error_message, O_message_type, O_message, O_wh, O_store, O_item)

The retrieve interface is a publicly exposed procedure and typically called by a RIB publication adaptor. Its parameters are well defined and arranged in a specific order. The message type is the RIB-defined "CustSaleCre", the message is the xml message, and the warehouse, store, and item are the key values for the message. Status code is one of four values, as shown in the following table. For more discussion of the status codes, refer to the Error Handling Guidelines document or the process flow in the following section. These codes come from an EAI team defined RIB_CODES package.

Code	Meaning
MFM_FATAL_ERROR	A fatal error occurred in the procedure.
MFM_SUCCESS	A message was retrieved.
MFM_NO_MSG	No message was retrieved.
MFM_WARNING	The next message on the queue cannot be published because a related family member with a lower sequence number is in error status.

The error text parameter contains application-generated information, such as the application's sequence number of the message that failed, and the Oracle or other error that occurred when the retrieval failed.

Message family queue

RIB_COSALE_MFQUEUE

The message family queue is a table with the following structure:

Column Name	Data Type/Length	Required	Constraints, Indices, Primary Keys	Description
SEQ_NO	NUMBER(15)	YES	PK, I1	sequence number of message in family
PUB_STATUS	VARCHAR2(1)	YES	(U, W, E, S), I1	publication statuses: unpublished, warning, error, skip
MESSAGE_TYPE	VARCHAR2(15)	YES		message type
MESSAGE	CLOB	YES		xml message
WH	VARCHAR2(12)	YES		Key value
STORE	VARCHAR2(10)	YES		key value
ITEM	VARCHAR2(25)	YES		key value

Message family sequence

RIB_COSALE_MFQUEUE_S

Each message family has its own sequence that generates sequence numbers for the population of the message family queue's seq_no column.

Process flow

addtoq

- Addtoq inserts a new message into the table rib_cosale_mfqueue. It gets a new sequence number from the rib_cosale_mfqueue_s sequence. The message_type is “CustSaleCre” and pub_status is set to “U” for unpublished.

getnxt

- 1 Retrieve a row from the RIB_COSALE_MFQUEUE corresponding to the lowest sequence number on the table that is in a “U”.
- 2 Removes the selected row from the table, returning the message, message_type, warehouse, store, and item via the procedure’s output parameters, with a status code of MFM_SUCCESS.
- 3 If the procedure encounters an error it updates the status of the message to E and returns the MFM_FATAL_ERROR status code. Error information, such as the application’s message sequence number and possible Oracle errors should be returned in the error text.
- 4 If the next message on the queue cannot be published because a related family member with a lower sequence number is in error status, a status code of MFM_WARNING is returned.
- 5 If there are no messages to retrieve, a MFM_NO_MSG status code is returned.