

Retek[®] Integration Bus[™] 11.0

Performance Tuning Guide

Corporate Headquarters:

Retek Inc.
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403
USA

888.61.RETEK (toll free US)
Switchboard:
+1 612 587 5000

Fax:
+1 612 587 5100

European Headquarters:

Retek
110 Wigmore Street
London
W1U 3RW
United Kingdom

Switchboard:
+44 (0)20 7563 4600

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

Fax:
+44 (0)20 7563 46 10

The software described in this documentation is furnished under a license agreement, is the confidential information of Retek Inc., 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, and the copyright notice may not be removed without the consent of Retek Inc.

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.

Retek[®] Integration Bus[™] is a trademark of Retek Inc.

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

This unpublished work is protected by confidentiality agreement, and by trade secret, copyright, and other laws. In the event of publication, the following notice shall apply:

©2004 Retek Inc. All rights reserved.

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.

Customer Support

Customer Support hours

Customer Support is available 7x24x365 via email, phone, and Web access.

Depending on the Support option chosen by a particular client (Standard, Plus, or Premium), the times that certain services are delivered may be restricted. Severity 1 (Critical) issues are addressed on a 7x24 basis and receive continuous attention until resolved, for all clients on active maintenance. Retek customers on active maintenance agreements may contact a global Customer Support representative in accordance with contract terms in one of the following ways.

Contact Method Contact Information

E-mail support@retек.com

Internet (ROCS) rocs.retek.com
Retek's secure client Web site to update and view issues

Phone +1 612 587 5800

Toll free alternatives are also available in various regions of the world:

Australia	+1 800 555 923 (AU-Telstra) or +1 800 000 562 (AU-Optus)
France	0800 90 91 66
Hong Kong	800 96 4262
Korea	00 308 13 1342
United Kingdom	0800 917 2863
United States	+1 800 61 RETEK or 800 617 3835

Mail Retek Customer Support
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403

When contacting Customer Support, please provide:

- Product version and program/module name.
- Functional and technical description of the problem (include business impact).
- Detailed step-by-step instructions to recreate.
- Exact error message received.
- Screen shots of each step you take.

Contents

Purpose of this Document.....	1
General Performance Considerations	3
eGate Server Tuning	3
eGate Server	3
Retek RIB code.....	4
eWays and TAFRs.....	4
eWay Threading	4
Two-Phase Commit.....	4
Custom Publishers/Subscribers	5
Custom Publisher Tuning	5
Table Changes	5
Driving Cursor.....	5
Messages with Details	6
Threading	6
Custom Subscriber Tuning	6
Process Multiple Rows	6
Cache Variables.....	6
Threading	6

Purpose of this Document

The purpose of this document is to give general guidelines for configuring an environment for the optimal performance of the Retek Integration Bus (RIB) software. This document will highlight specific information about hardware and operating system software configurations.

Please refer to the *Retek 11.0 Software Performance Tuning Guide* for general performance tuning guidelines.

General Performance Considerations

There are three primary components to the RIB: the SeeBeyond eGate server and the Retek code that executes on that server, the publisher code that creates messages and publishes them to the RIB from the source system, and the subscriber code that consumes messages from the RIB and updates the destination system.

eGate Server Tuning

There are two primary parts to the eGate server that can be tuned. The eGate server itself is the first part which would include the Java Virtual Machine (JVM) that it is running on the server as well as the Java Message Service (JMS) configuration. The second part that can be tuned is the Retek RIB code running on the eGate server. This would include any TAFRs (Transport, Address, Format, and Route) and eWays.

eGate Server

The eGate server runs on a JVM and uses a JMS to hold the messages it is processing.

Java Virtual Machine

The eGate server ships with default JVM settings that should be sufficient for most customers.

Java Message Service

There are a couple of things that can be changed with the JMS. The segment size can be adjusted to more closely reflect the size of the messages being stored. This will eliminate wasted space and improve efficiency. The timetolive setting determines how long a message can remain on the JMS. Setting it to too short of a time will result in messages being destroyed before they are consumed. Setting it to be too long could result in a larger than necessary JMS. It should be adjusted to an appropriate value based on testing real messages and throughput.

The performance of the JMS can also be affected by the performance of the IO subsystem. Putting the JMS files on fast disk will improve performance.

Journaling

Journaling can be turned on for testing and debugging. It should be turned off in production because it has a significant negative performance impact.

Retek RIB code

The Retek RIB code is code running on the eGate server that was written by Retek.

eWays and TAFRs

The eWays and TAFRs run inside the JVMs. During configuration, memory settings should be set as well as turning the just in time compiler “on”. Through on-site testing, the memory setting should be adjusted so that garbage collection is rarely needed or occurs as rarely as possible.

eWay Threading

Each eWay runs in its own JVM and is configured separately. Multiple Java threads can be set up to run within the eWay to handle multiple threads of publishing and subscribing. The maximum number of threads before performance starts to degrade is four. If more threads are needed, another eWay for the same collaboration can be set up with a different name with up to four Java threads. The threading of eWays varies on a case-by-case basis so testing must be done to determine the correct configuration.

Two-Phase Commit

Two-phase commit is a relatively costly process that is used to ensure transactions complete for all sending and receiving sources. Reducing the number of two-phase commits will improve performance but it must be balanced against ensuring the process is reliable. Two-phase commits can be reduced by eliminating unnecessary data sources in the transactions. They can also be reduced by aggregating messages together to create larger, bulk type transactions. The outcome of two-phase commit varies on a case-by-case basis so testing must be done to determine the correct configuration.

Custom Publishers/Subscribers

Custom Publisher Tuning

Custom publishers can be tuned by making changes to the tables or by making changes to the PL/SQL packages and triggers.

Table Changes

The primary changes that can be made to the tables are to provide the appropriate indexes. The indexes normally used on the mfqueue tables are shown in Table 1.

Table 1: Indexes

Index	Column	Unique
pk	seq_no	Yes
u1	thread_no	Yes
u1	seq_no	Yes
il	<business object>	No

Driving Cursor

With the correct indexes in place, the driving cursor needs to be written to take advantage of them. It should be similar to the following code:

```

cursor C_QUEUE is
  select q.field1,
         q.field2,
         ...
         q.fieldx,
         q.seq_no,
         q.rowid
  from <<family>>_mfqueue q
 where q.seq_no = (select min(q2.seq_no)
                  from <<family>>_mfqueue q2
                  where q2.thread_no = I_thread_val
                  and q2.pub_status = 'U'
                  and q2.seq_no > L_seq_limit)
 and q.thread_no = I_thread_val;

```

Messages with Details

When publishing messages with details, bulk collect should be used whenever possible.

Threading

The Retek Integration Bus has had little trouble handling very large volumes of messages. The issue generally is being able to publish the messages fast enough to keep the mfqueue tables clear. All high volume message families should have publishers set up to allow for threading. The correct number of threads of each publisher varies on a case-by-case basis and must be determined through testing at each customer site.

Custom Subscriber Tuning

The subscribers can be tuned by making changes to process multiple rows in a single transaction and to cache variables.

Process Multiple Rows

An init function should initialize global tables that will store the data for bulk binding. As the message is processed, the data in the detailed nodes should be put into the global tables. When all the detailed nodes are processed, a flush function should do a bulk insert/update of the data using a forall statement. This process will significantly improve the throughput for consuming messages. It does come at a cost though; while the data is being processed, row locks will be created. These locks remain in place until all the data is processed. The potential cost is that these locks could block messages in other message families from being consumed.

Cache Variables

Values that are frequently used can be cached so that they do not have to be queried each time they are needed. The time spent executing an individual statement may not be high, but executing it thousands of times does add up. The values can be stored in global variables. An example of this would be selecting inventory status codes and putting them in a global table. Those values will not change during a single database session.

Threading

Again, the Retek Integration Bus has had little trouble handling very large volumes of messages. All high volume message families should have subscribers set up to allow for threading. The correct number of threads of each subscriber varies on a case-by-case basis and must be determined through testing at each customer site.