HP Reliable Transaction Router

Release Notes

September 2009

HP Reliable Transaction Router (RTR) is fault tolerant transactional messaging middleware used to implement large, multitier, distributed applications using client/server technology.

These notes describe the new features, corrections, restrictions, and known problems with workarounds for RTR Version 5.2.

Software Version: HP Reliable Transaction Router Version 5.2

Hewlett-Packard Company
Palo Alto, California

© Copyright 2009 Hewlett-Packard Development Company, L.P.

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Microsoft and Windows are U.S. registered trademarks of Microsoft Corporation.

Intel and Itanium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group. Java is a US trademark of Sun Microsystems, Inc.

This document was prepared using DECdocument, Version 3.3-1B.

Contents

Preface v

1 RTR Version 5.2

1.1 New Features 1–1
1.2 Enhancements 1–2
1.3 Problems Fixed 1–3

Tables

1 RTR Documents v

Preface

Purpose of these Release Notes

This document provides information for RTR Version 5.2.

Intended Audience

These Release Notes are intended for all users of RTR. Please read this entire document before using the product.

Related Documentation

Table 1 describes RTR documents and groups them by audience.

Table 1 RTR Documents

Document	Content
For all users:	
HP Reliable Transaction Router Release Notes[1]	Describes new features, corrections, restrictions, and known problems for RTR.
HP Reliable Transaction Router Getting Started	Provides an overview of RTR technology and solutions, and includes the glossary that defines all RTR terms.
HP Reliable Transaction Router *Software Product Description*	Describes product features.
For the system manager:	
HP Reliable Transaction Router Installation Guide	Describes how to install RTR on all supported platforms.
HP Reliable Transaction Router System Manager's Manual	Describes how to configure, manage, and monitor RTR.
For the application programmer:	
HP Reliable Transaction Router Application Design Guide	Describes how to design application programs for use with RTR, with both C++ and C interfaces.
HP Reliable Transaction Router JRTR Getting Started [2]	Provides an overview of the object-oriented JRTR Toolkit including installation, configuration and Java programming concepts, with links to additional online documentation.
HP Reliable Transaction Router C++ Foundation Classes	Describes the object-oriented C++ interface that can be used to implement RTR object-oriented applications.

[1]Distributed on software kit.

[2]In downloadable kit.

(continued on next page)

Table 1 (Cont.) RTR Documents

Document	Content
HP Reliable Transaction Router C Application Programmer's Reference Manual	Explains how to design and code RTR applications using the C programming language and the RTR C API. Contains full descriptions of the basic RTR API calls.

You can find additional information about RTR at the following World Wide Web address:

`http://www.hp.com/go/rtr`

Conventions

The following conventions are used in this manual:

Convention	Meaning
italic text	Italic text emphasizes important information, indicates variables, and indicates complete titles of manuals. Italic text also represents information that can vary in system messages (for example, Internal error *number*), command lines (for example, /PRODUCER=*name*), and command parameters in text.
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
`user input`	This bold typeface is used in interactive examples to indicate typed user input.
`system output`	This typeface is used in interactive and code examples to indicate system output.

Reader's Comments

HP welcomes your comments on this manual. Please send comments to openvmsdoc@hp.com.

1
RTR Version 5.2

1.1 New Features

This section describes the new features added in this release.

- **Support for Linux 5.2 (x86_64) servers**

  RTR is now available on HP Proliant x86 64-bit Linux platform.

  RedHat Cluster-suite and Global File System (GFS) are required for configuring RTR in an ACTIVE/STANDBY mode.

  - Platform Support Details:

    Kernel Version: Redhat Linux 2.6.18-92.el5

    Linux OS Version: RHEL 5.2

  - Compiler Details:

    C Compiler: gcc version 4.1.2 20071124 (Red Hat 4.1.2-42)

  - RTR application compiler flags

    Compile RTR applications with fPIC and m64 flag of gcc compiler.

Note

HP RTR for Linux 5.2 (x86_64) supports only 64-bit applications.

Note

While configuring RTR in the ACTIVE/STANDBY mode, the journal must be created in the shared disk (GFS), which is accessible from all nodes configured in the Linux cluster.

For RTR, to work in the ACTIVE/STANDBY mode, set the following environmental variables:

- RTR_CLU_MEM_NODES: This variable enables RTR to recognize the back-end nodes of the cluster that is participating in the ACTIVE/STANDBY mode. For example:

  ```
  RTR_CLU_MEM_NODES="BE1, BE2"
  ```

- RTR_CLU_LCK_DISK: This variable enables RTR to maintain the "cluster node locks" on a common shared disk in the cluster. All back-end nodes in the cluster must use the same disk for maintaining "cluster node locks". For example:

  ```
  RTR_CLU_LCK_DISK="/dev/mapper/rtrvg-lvol0"
  ```

- **Support for Microsoft Windows x86_64**

  RTR (32-bit) is now qualified on Windows x86_64 operating system.

- **Selective data compression**

  In an RTR configuration comprising of several systems with transaction routers and back-end nodes, the network must have sufficient bandwidth to carry the RTR management messages and application payloads. Insufficient network bandwidth can result in RTR buffering the data eventually leading to its failure because of the non-availability of memory.

  To overcome this problem, RTR is enhanced to selectively compress the data packets to reduce the load on the network. This feature is available through an environment variable, `RTR_COMPRESS_PCKT`. By default, this feature is disabled. To enable data compression, set `RTR_COMPRESS_PCKT` to 1.

  To increase the efficiency of compression, you can also use the environment variable, `RTR_COMPRESS_PCKT_THRESHOLD`, and select the minimum size of the packet that has to be compressed. The packets with size greater than or equal to `RTR_COMPRESS_PCKT_THRESHOLD` will be compressed and smaller packets will not be compressed.

- **Support for 64-bit library on OpenVMS**

  RTR is enhanced to extend the memory allocation capacity of RTR processes by supporting 64-bit pointers.

  To make use of the 64-bit pointers, the applications must be recompiled with the `pointer_size=64` qualifier of the C/C++ compiler.

  The 32-bit applications that use V3/V2 API does not require recompilation or relinking to work with the 64-bit RTR library. An existing application, based on V3/V2 API, which is compiled with a 32-bit and linked with an old RTR library can work seamlessly with the 64-bit library.

  ### Restriction

  Applications that use V2 API cannot be compiled with the `pointer_size=64` qualifier because RTR does not support this API in the 64-bit format. Applications must migrate to V3 API to make use of the 64-bit pointers on OpenVMS.

1.2 Enhancements

This section describes the enhancements made in this release.

- **Enhanced error messages**

  Selective RTR error messages are enhanced to display the appropriate error or information, along with the cause for the error. You can find the error messages in the RTR log file.

  RTR online help is also enhanced to display the detailed explanation for various error codes.

- **Monitor Picture enhancement**

  A new monitor picture is added that displays the front-end links on the transaction router node. When the `MONITOR` command is executed on a transaction router node, the monitor picture displays all the front-ends currently connected to the router. The `MONITOR` command for trtofe.mon supports a new qualifier `PERFACILITY`, which displays the connected front-ends at the facility level.

For example,

```
$ RTR MONITOR TRTOFE /PERFACILITY=TEST
```

1.3 Problems Fixed

This section describes the problems that are fixed in this release:

- QUIX QXCM1000947175/ WFM 4600588231, RTRACP on primary backend crashes due to maximum utilization of memory.

  RTRACP temporarily stores messages in its internal buffers until the messages are successfully delivered to the network stack for actual transportation to a remote node. In an event of network saturation, RTR might take more time to delete the messages from its buffers due to congestion at network level. In a large configuration comprising several routers and backends, this phenomena leads to pile up of messages and crashes due to maximum utilization of memory.

  To control the message pile up in RTRACP process, a new environment variable, RTR_NIO_BLOCKAGE is introduced, which specifies the maximum duration of the packet to be held in its internal buffers. After the specified time period, it discards the message and also disconnects the link associated with that buffer, which will be restored automatically by RTR later. The minimum, maximum and default value of this environment variable is 15, 120 and 30 seconds respectively.

- QUIX QXCM1000928829/ WFM 2607634652, RTR loops in the listener socket (OpenVMS only)

  In a rare scenario, RTR loops in the listener socket and constantly logs the "accept" error message in its log file. As a result, this leads to the high utilization of the CPU by the ACP process.

  This problem is corrected in this release.

- When the `RTR$SNAPSHOT` command procedure is executed, the RTRCLI crashes with an ACCVIO error.

  This problem is fixed in this release.

- In a rare failover condition, the back-end transactions in the DONE state are not cleaned up because of the pending "forget" message from the transaction router.

  This problem is fixed in this release.

- A combination of read-write and read-only transactions processed by a strict order shadow partition at the primary site leads to transaction pile up at the secondary site. The primary ACP incrementing the commit sequence number of the read-only transactions never gets processed at the secondary ACP. Because the criteria used to maintain the ordering of messages at secondary ACP fails, all transactions at secondary ACP is stored in its buffer.

  This problem is fixed in this release.

- RTRACP obtains only the primary interface address when there are multiple interfaces configured with Internet Protocol on a Linux (Integrity servers and x86_64) system. This results in RTR reporting the LNXINETNOIP error in its log file.

  This problem is fixed in this release.

RTR Version 5.2
1.3 Problems Fixed

- QUIX QXCM1000853085/WFM 1601181965-163, Front-ends are not able to connect to RTR Daemon (RTRD)

  RTRD is now modified to wait for a user specified time period to complete the processed ACP port lookup or web service request. If RTRD cannot complete the service request in the given time period, it aborts the request and proceeds with a new service request.

  This mechanism provides additional protection to RTRD, which prevents it from blocking on a service request indefinitely. When RTRD aborts the request due to the timeout, an error message is created in the log file.

  By default, the time period to complete a service request is set to 10 seconds. This time period (in seconds) is configurable using the logical, RTR_RTRD_TIMOUT, which accepts the values between 1 and 60 seconds range. The minimum value of the time period is 1 second and the maximum value is 60 seconds.

  On OpenVMS, the user account from which RTRD process is started must have an appropriate TQELm quota.

  On a router node, the TQElm quota must be set to twice the number of front-ends and back-ends or 2000 or whichever is higher. The TQElm quota for a back-end node must be set to twice the number of routers or 2000 or whichever is higher.

  The TQElm quota requirement of ACP and Com-server processes remains unchanged because of this update to RTRD. For information about starting ACP with the appropriate command-line parameters related to process quota, see the *HP Reliable Transaction Router System Manager's Manual*.

- The `RTR SHOW PROCESS` command output displays the process name followed by the user name on HP-UX, Linux, and Windows platforms. In the earlier versions of RTR, this format of output was displayed only on OpenVMS.