



Ultra Messaging (Version 6.12)

Release Notes

Copyright (C) 2004-2017, Informatica Corporation. All Rights Reserved.

Contents

1	Introduction	5
2	UM Version 6.12	7
2.1	Enhancements for 6.12	7
2.1.1	Streaming Enhancements for 6.12	7
2.1.2	Persistence Enhancements for 6.12	8
2.1.3	Queuing Enhancements for 6.12	8
2.1.4	Dynamic Router Enhancements for 6.12	8
2.2	Fixed Limitations for 6.12	8
2.2.1	Streaming Fixed Limitations for 6.12	8
2.2.2	Persistence Fixed Limitations for 6.12	9
2.2.3	Queuing Fixed Limitations for 6.12	9
2.2.4	Dynamic Router Fixed Limitations for 6.12	10
3	Known Limitations	11
3.1	Streaming Known Limitations	11
3.2	Persistence Known Limitations	12
3.3	Queuing Known Limitations	14
3.4	Dynamic Router Known Limitations	14
4	Upgrade Procedure	17
4.1	Default Values for Configuration Options	17
4.2	Deprecations	17
4.3	Upgrading Brokered Queuing and JMS	18
4.4	Upgrading From Version Pre-6.0	18
4.4.1	Log Messages During Upgrade	19
5	UM Version 6.11.1	21
5.1	Enhancements for 6.11.1	21
5.1.1	Streaming Enhancements for 6.11.1	21
5.1.2	Persistence Enhancements for 6.11.1	21
5.1.3	Queuing Enhancements for 6.11.1	22
5.1.4	Dynamic Router Enhancements for 6.11.1	22

5.2	Fixed Limitations for 6.11.1	22
5.2.1	Streaming Fixed Limitations for 6.11.1	22
5.2.2	Persistence Fixed Limitations for 6.11.1	22
5.2.3	Queuing Fixed Limitations for 6.11.1	23
5.2.4	Dynamic Router Fixed Limitations for 6.11.1	23
5.3	Deprecations for 6.11.1	23
6	UM Version 6.11	25
6.1	Enhancements for 6.11	25
6.1.1	Streaming Enhancements for 6.11	25
6.1.2	Persistence Enhancements for 6.11	25
6.1.3	Queuing Enhancements for 6.11	26
6.1.4	Dynamic Router Enhancements for 6.11	26
6.2	Fixed Limitations for 6.11	26
6.2.1	Streaming Fixed Limitations for 6.11	26
6.2.2	Persistence Fixed Limitations for 6.11	28
6.2.3	Queuing Fixed Limitations for 6.11	29
6.2.4	Dynamic Router Fixed Limitations for 6.11	29
7	UM Version 6.10.1	31
7.1	Enhancements for 6.10.1	31
7.1.1	Streaming Enhancements for 6.10.1	31
7.1.2	Persistence Enhancements for 6.10.1	31
7.1.3	Queuing Enhancements for 6.10.1	31
7.1.4	Dynamic Router Enhancements for 6.10.1	32
7.2	Fixed Limitations for 6.10.1	32
7.2.1	Streaming Fixed Limitations for 6.10.1	32
7.2.2	Persistence Fixed Limitations for 6.10.1	32
7.2.3	Queuing Fixed Limitations for 6.10.1	33
7.2.4	Dynamic Router Fixed Limitations for 6.10.1	33
8	UM Version 6.10	35
8.1	Enhancements for 6.10	35
8.1.1	Streaming Enhancements for 6.10	35
8.1.2	Persistence Enhancements for 6.10	38
8.1.3	Queuing Enhancements for 6.10	38
8.1.4	Dynamic Router Enhancements for 6.10	38
8.2	Fixed Limitations for 6.10	38
8.2.1	Streaming Fixed Limitations for 6.10	39
8.2.2	Persistence Fixed Limitations for 6.10	40
8.2.3	Queuing Fixed Limitations for 6.10	41

8.2.4	Dynamic Router Fixed Limitations for 6.10	41
8.3	Deprecations for 6.10	42
8.3.1	Streaming Deprecations for 6.10	42
8.3.2	Persistence Deprecations for 6.10	42
8.3.3	Dynamic Router Deprecations for 6.10	42
9	UM Version 6.9.2	43
9.1	Enhancements for 6.9.2	43
9.1.1	Streaming Enhancements for 6.9.2	43
9.1.2	Persistence Enhancements for 6.9.2	43
9.1.3	Queuing Enhancements for 6.9.2	43
9.1.4	Dynamic Router Enhancements for 6.9.2	44
9.2	Fixed Limitations for 6.9.2	44
9.2.1	Streaming Fixed Limitations for 6.9.2	44
9.2.2	Persistence Fixed Limitations for 6.9.2	44
9.2.3	Queuing Fixed Limitations for 6.9.2	44
9.2.4	Dynamic Router Fixed Limitations for 6.9.2	45
10	UM Version 6.9.1	47
10.1	Enhancements for 6.9.1	47
10.1.1	Streaming Enhancements for 6.9.1	47
10.1.2	Persistence Enhancements for 6.9.1	47
10.1.3	Queuing Enhancements for 6.9.1	48
10.1.4	Dynamic Router Enhancements for 6.9.1	48
10.2	Fixed Limitations for 6.9.1	48
10.2.1	Streaming Fixed Limitations for 6.9.1	48
10.2.2	Persistence Fixed Limitations for 6.9.1	48
10.2.3	Queuing Fixed Limitations for 6.9.1	49
10.2.4	Dynamic Router Fixed Limitations for 6.9.1	49
11	UM Version 6.9	51
11.1	Enhancements for 6.9	51
11.1.1	Streaming Enhancements for 6.9	51
11.1.2	Persistence Enhancements for 6.9	52
11.1.3	Queuing Enhancements for 6.9	52
11.1.4	Dynamic Router Enhancements for 6.9	52
11.2	Fixed Limitations for 6.9	52
11.2.1	Streaming Fixed Limitations for 6.9	52
11.2.2	Persistence Fixed Limitations for 6.9	53
11.2.3	Queuing Fixed Limitations for 6.9	53
11.2.4	Dynamic Router Fixed Limitations for 6.9	54

12 UM Version 6.8	55
12.1 Enhancements for 6.8	55
12.1.1 Streaming Enhancements for 6.8	55
12.1.2 Persistence Enhancements for 6.8	56
12.1.3 Queuing Enhancements for 6.8	59
12.1.4 Dynamic Router Enhancements for 6.8	59
12.2 Fixed Limitations for 6.8	60
12.2.1 Streaming Fixed Limitations for 6.8	60
12.2.2 Persistence Fixed Limitations for 6.8	62
12.2.3 Queuing Fixed Limitations for 6.8	63
12.2.4 Dynamic Router Fixed Limitations for 6.8	64
12.3 Deprecations for 6.8	64
12.3.1 Persistence Deprecations for 6.8	65
12.3.2 Queuing Deprecations for 6.8	65
12.3.3 Dynamic Router Deprecations for 6.8	65
13 UM Version 6.7.3	67
13.1 Enhancements for 6.7.3	67
13.1.1 Streaming Enhancements for 6.7.3	67
13.1.2 Persistence Enhancements for 6.7.3	67
13.1.3 Queuing Enhancements for 6.7.3	68
13.1.4 Dynamic Router Enhancements for 6.7.3	68
13.2 Fixed Limitations for 6.7.3	68
13.2.1 Streaming Fixed Limitations for 6.7.3	68
13.2.2 Persistence Fixed Limitations for 6.7.3	68
13.2.3 Queuing Fixed Limitations for 6.7.3	69
13.2.4 Dynamic Router Fixed Limitations for 6.7.3	69
14 UM Version 6.7.2	71
14.1 Enhancements for 6.7.2	71
14.1.1 Streaming Enhancements for 6.7.2	71
15 UM Version 6.7.1	73
15.1 Enhancements for 6.7.1	73
15.1.1 Streaming Enhancements for 6.7.1	73
15.1.2 Persistence Enhancements for 6.7.1	73
15.1.3 Queuing Enhancements for 6.7.1	73
15.1.4 Dynamic Router Enhancements for 6.7.1	74
15.2 Fixed Limitations for 6.7.1	74
15.2.1 Streaming Fixed Limitations for 6.7.1	74
15.2.2 Persistence Fixed Limitations for 6.7.1	75

15.2.3	Queuing Fixed Limitations for 6.7.1	76
15.2.4	Dynamic Router Fixed Limitations for 6.7.1	76
16	UM Version 6.7	79
16.1	Enhancements for 6.7	79
16.1.1	Streaming Enhancements for 6.7	79
16.1.2	Persistence Enhancements for 6.7	80
16.1.3	Queuing Enhancements for 6.7	80
16.1.4	Dynamic Router Enhancements for 6.7	80
16.2	Fixed Limitations for 6.7	81
16.2.1	Streaming Fixed Limitations for 6.7	81
16.2.2	Persistence Fixed Limitations for 6.7	83
16.2.3	Queuing Fixed Limitations for 6.7	85
16.2.4	Dynamic Router Fixed Limitations for 6.7	85
16.3	Deprecations for 6.7	86
16.3.1	Streaming Deprecations for 6.7	86
16.3.2	Persistence Deprecations for 6.7	86
17	UM Version 6.5	89
17.1	Enhancements for 6.5	89
17.1.1	Streaming Enhancements for 6.5	89
17.1.2	Persistence Enhancements for 6.5	89
17.1.3	Queuing Enhancements for 6.5	90
17.1.4	Dynamic Router Enhancements for 6.5	90
17.2	Ultra Messaging System Monitoring Option	90
17.3	Fixed Limitations for 6.5	91
17.3.1	Streaming Fixed Limitations for 6.5	91
17.3.2	Persistence Fixed Limitations for 6.5	91
17.3.3	Queuing Fixed Limitations for 6.5	91
17.3.4	Dynamic Router Fixed Limitations for 6.5	91
18	UM Version 6.1	93
18.1	Enhancements for 6.1	93
18.1.1	Streaming Enhancements for 6.1	93
18.1.2	New Transport LBT-SMX	94
18.1.3	Persistence Enhancements for 6.1	94
18.1.4	Queuing Enhancements for 6.1	95
18.1.5	Dynamic Router Enhancements for 6.1	95
18.2	Fixed Limitations for 6.1	95
18.2.1	Streaming Fixed Limitations for 6.1	95
18.2.2	Persistence Fixed Limitations for 6.1	95

18.2.3	Queuing Fixed Limitations for 6.1	96
18.2.4	Dynamic Router Fixed Limitations for 6.1	96
19	UM Version 6.0	97
19.1	Enhancements for 6.0	97
19.1.1	Streaming Enhancements for 6.0	97
19.1.2	Persistence Enhancements for 6.0	98
19.1.3	Queuing Enhancements for 6.0	99
19.1.4	Dynamic Router Enhancements for 6.0	100
19.2	Fixed Limitations for 6.0	100
19.2.1	Streaming Fixed Limitations for 6.0	100
19.2.2	Persistence Fixed Limitations for 6.0	101
19.2.3	Queuing Fixed Limitations for 6.0	101
19.2.4	Dynamic Router Fixed Limitations for 6.0	102
19.3	Deprecations for 6.0	102
19.3.1	Streaming Deprecations for 6.0	102

Chapter 1

Introduction

This document describes the important changes made to the Ultra Messaging product family with each release. The document is cumulative across the major version number, and also across the core Ultra Messaging products: Streaming Edition (UMS), Persistence Edition (UMP), Queuing Edition (UMQ), and the Dynamic Routing Option (DRO).

Attention

See the [Documentation Introduction](#) for important information on copyright, patents, information resources (including Knowledge Base, and How To articles), Marketplace, Support, and other information about Informatica and its products.

See **UM Glossary** for Ultra Messaging terminology, abbreviations, and acronyms.

Chapter 2

UM Version 6.12

2.1 Enhancements for 6.12

2.1.1 Streaming Enhancements for 6.12

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- **TCP-Based Topic Resolution.** 6.12 introduces a new component, the SRS (Stateful Resolution Service). This is the first phase of a plan to significantly improve UM's topic resolution characteristics. For more information, see **Topic Resolution Description**.
- **Streaming Receiver Optimization.** Message receive latency has been improved with the following features:
 - **Receive Buffer Recycling**
 - **Single Receiving Thread**
 - **Context Lock Reduction**
 - **Gettimeofday Reduction**
- **Smart Source Message Sizing Flexibility.** The **Smart Sources** feature now supports greater flexibility in application message sizes, including:
 - UM Message Fragmentation.
 - User Supplied Buffers.

For more information, see **Smart Source Message Buffers**.

- **Windows Service Enhancements.** Support for running UM Daemons as Windows Services has been improved. This includes the UM Message Router (DRO), the persistent Store, and the unicast topic resolution daemon (lbmrdr). For more information, see **UM Daemons as Windows Services**.
- **XSP Enhancements.** The **Transport Services Provider (XSP)** feature has been enhanced to support the following new capabilities:
 - **Persistence.** Persistent receivers can now be assigned to XSP.

- Registration of File Descriptors. The **Transport Services Provider (XSP)** feature has been enhanced to support registration of application file descriptors. For more information, see **Other XSP Operations**.
- **Ultra Load Balancing Throughput.** The **Ultra Load Balancing (ULB)** feature has been enhanced to significantly improve throughput when used with unicast transport protocols TCP and LBT-RU. For more information see **ULB Performance**.

2.1.2 Persistence Enhancements for 6.12

The following new features and enhancements apply to UMP and UMQ products.

- None.

2.1.3 Queuing Enhancements for 6.12

The following new features and enhancements apply to the UMQ product.

- None.

2.1.4 Dynamic Router Enhancements for 6.12

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- None.

2.2 Fixed Limitations for 6.12

2.2.1 Streaming Fixed Limitations for 6.12

The following bug fixes apply to UMS, UMP, and UMQ products.

3398	FIXED: <i>Windows-based UM closes TCP connections abruptly, causing TCP "RST" packets on the network, and "connection reset" warnings in UM log files.</i> Windows-based UM now closes TCP connections gracefully.
------	---

10518	FIXED: <i>When installing UM on a Windows machine using the UM installer, the system PATH can be corrupted or cleared if the updated length is greater than 1023 bytes.</i> This is related to a limitation in the Windows installer that UM uses; it fails on long PATH content. UM had a bug in which UM did not detect the failure and instead wrote the bad PATH. UM has been updated to detect the failure, issue an error, and not update the PATH.
10380, 10381	FIXED: <i>lbm_rcv_delete_ex and lbm_wildcard_rcv_delete_ex can crash when unsubscribing from spectrum channels.</i>
10469, 10486, 10466	FIXED: <i>When sending a message using a Smart Source send API, UM can crash with a failed assertion of the form: "failed assertion [...->sminfo.timer->_id == -1]". This happens when a TSNI is in the process of being sent by the context thread at the same time that the application sends a message.</i>
10492	FIXED: <i>When the lbmrd is configured for incorrect interface specification, either in its XML configuration file, or on the command line, the lbmrd did not detect any error and appears to be running. However, it does not function. Also, the lbmrd does accept interface specifications other than a simple IP address.</i> The lbmrd interface can now be specified as a simple IP addresses, a CIDR network specification (e.g. 10.1.2.0/24), a quoted device name (e.g. "eth0"), or a DNS host name. The lbmrd will report an error if an invalid interface specification is configured.
10288	FIXED: <i>Sending Unicast Immediate Requests on Sparc machines can result in high CPU burn with no messages being sent.</i>

2.2.2 Persistence Fixed Limitations for 6.12

The following bug fixes apply to UMP and UMQ products.

10446, 10450	FIXED: <i>When applications and/or Stores are restarted in certain orders, the receiver can log the warning, "Core-9743-08: WARNING: proactive keepalive is not supported at the store". This can happen even though the Store does support the Proactive Keepalive feature.</i>
10379	FIXED: <i>When a persistent source deregisters from the Store by API call (lbm_src_ume_↵deregister()), the Store will occasionally crash.</i>

2.2.3 Queuing Fixed Limitations for 6.12

The following bug fixes apply to the UMQ product.

9974	FIXED: <i>There was a scenario where a late arriving ACK from a ULB receiver could cause the ULB source to segmentation fault inside lbm_ulb_src_ctlr_handle_ack().</i>
------	---

2.2.4 Dynamic Router Fixed Limitations for 6.12

The following bug fixes apply to the Dynamic Routing Option (DRO).

10443	<p>FIXED: <i>When a DRO portal is queuing outgoing messages (perhaps due to a low rate limiter or a low-bandwidth link), and a topic has more than delivery_control_maximum_burst_loss (receiver) (default 1024) messages queued to be sent, there is a possibility that receivers will deliver a burst loss event to the application and not deliver the queued messages, even though they are eventually sent. This is because the originating source sent a TSNI, and the DRO allowed it to bypass the data queue and be sent before the queued data messages. Thus the receiver sees a gap greater than the burst loss configuration and does not wait for the missing data to be sent.</i></p> <p>The DRO now queues TSNI behind data messages.</p>
-------	---

Chapter 3

Known Limitations

3.1 Streaming Known Limitations

The following open limitations apply to UMS, UMP, and UMQ products.

Change Request	Description
10288	Starting in UM 6.7, setting any flags with the lbm_unicast_immediate_request() API can cause undefined behavior.
10136	When an XML file is used to configure UM, an invalid configuration option or value logs appropriate error messages, but does not return failure status to the application. As a workaround, the application logger callback can check for logged messages that start with "Core-5626-" and trigger their own error handler. Note that setting up the logger callback has a clientd (see lbm_log()), so a shared structure can be established for the logger callback to communicate any matches it finds to the application thread calling lbm_config() .
9964	The use of Doxygen to generate the .NET documentation has resulted in the omission of the "LBMCS.chm" windows help package.
9718	The encryption feature does not implement elliptic curve encryption. Cipher suites with "ECDH", "ECDHE", and "ECDSA" cannot be used. Use "DHE" and "RSA" instead.
4126	When using LBT-IPC transports, 64-bit applications cannot interoperate with 32-bit applications. This includes router endpoint portals.
6553, 6441	You cannot use Unicode PCRE characters in wildcard receiver patterns on any system that communicates with a HP-UX or AIX system.
6828, 6830, 6831	There are severe restrictions on the use of NAT with UM. Normally, a UM Router should be used with peer links to transit a NAT. However, there are very restrictive scenarios where Unicast Topic Resolution with lbmrtd can be used to establish direct communication across a NAT. See Network Address Translation (NAT) .
7762	When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions before 9.4 have not produced this behavior.

Change Request	Description
7992, 7984, 7990	The LBT-SMX transport does not support Topic Sequence Number Information (TSNI) messages. Thus, if LBT-SMX traffic routes through one or more UM Routers, and the original LBT-SMX source does not send messages regularly, remote receivers do not detect and notify the application of topic-level tail loss. Also, UM might not deliver midstream topic-level loss notifications in a timely manner. Remote receivers may also experience repeated transport disconnects and reconnects, and may miss Beginning Of Session (BOS) messages for the remote transport that carries data from the originating SMX source. Remote receivers may also experience head loss, similar to what would occur if the remote UM Router had started after the originating SMX source starts sending. When configuring SMX sources to send through a UM Router, ensure receiving applications do not rely upon these missing TSNI-related behaviors.
8042	It is possible for a thread deadlock to occur in the Java HotSpot Virtual Machine when initializing direct byte buffers. To avoid this bug, upgrade to Java 1.6.0_18 or later. For more information see http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6791815 .
8266	With Ultra Messaging on the AIX platform, you might encounter library linking problems due to an older version of 'libcrypto.a' in the Ultra Messaging core package. To avoid this, set the LIBPATH in such a way that it takes the system libcrypto.a ahead of the Ultra Messaging 'libcrypto.a'.
8727, 8603	You cannot use lbm_rcv_retrieve_transport_stats() to retrieve transport statistics from receivers with SMX sources. Instead, to retrieve these statistics use Automatic Monitoring.
8735, 8323	In AIX systems, you cannot send messages with a zero-length payload.
8744	If the lbmrd configuration file does not specify the version attribute within the '<lbmrd>' element, the daemon quits immediately without an error message. To resolve this issue, change the '<lbmrd>' element to '<lbmrd version="1.0">'.
8794, 8475	Configuring sources with LBT-RU, source-side filtering, and adaptive batching might cause a crash. Note that adaptive batching is deprecated and will be removed from a future release.
8820, 7368	Setting transport_tcp_activity_method (receiver) to 'SO_KEEPALIVE' without setting transport_tcp_activity_timeout (receiver) to a value other than the default value of 0 causes a receiver create fatal assert: 's_entry->topic.rcv!=NULL'. Also, any other configuration that causes a receiver create to fail triggers the fatal assert: 's_entry->topic.rcv!=NULL'.
9030, 8799	You cannot set transport options in-line for lbmmon using a UM XML configuration file. Instead, use a Plain Text configuration file to set these options.

3.2 Persistence Known Limitations

The following open limitations fixes apply to UMP and UMQ products.

Change Request	Description
10171	A set of default operational configuration options do not satisfy Persistence recommendations. Hung registrations can result. Persistence users are advised to follow the recommendations in Preventing Store Registration Hangs .

Change Request	Description
2982	Receivers using event queues and Spectrum with UMP can experience a SIGSEGV while shutting down if events still exist on the event queue when the application deletes the event queue. As a workaround, use LBM_EVQ_BLOCK when dispatching event queues. During application shutdown, call <code>lbm_evq_unblock()</code> after deleting receivers associated with the event queue, but before deleting any context objects. When the dispatch thread exits, you can continue with context deletion.
4543	When running a store on a Solaris machine, you may experience registration failures after a few minutes. The store repeatedly reports the error: [WARNING]: wait returned error 4 [select: (22) Invalid argument] Setting fd_management_type (context) to 'devpoll' prevents this problem.
4667	Multi-transport Threads do not support Persistence. Note that MTT is deprecated, and the new XSP feature provides improved multi-threading for receivers, and is compatible with Persistence. See Transport Services Provider (XSP) .
5841	A UMP store daemon, <code>umestored</code> , may shut down unexpectedly if you enable proxy sources and UM is not able to create a transport session for a needed proxy source. For example, if the store is configured to use TCP transports for proxy sources (the default), and transport_tcp_port_high (context) minus transport_tcp_port_low (context) is 8, then when the 9th proxy source is created, UM will be unable to assign a port for that transport session. This is because UM will attempt to create up to transport_tcp_maximum_ports (context) transport sessions, which defaults to 10. transport_tcp_port_high (context) minus transport_tcp_port_low (context) must be at least transport_tcp_maximum_ports (context) , and may need to be even larger if other applications and/or daemons on the same machines are configured to use ports from the same range. Informatica recommends configuring large port ranges appropriate to the transport type.
6843	An overly aggressive configuration on the source can cause the source to declare the store unresponsive while the source processes stability acknowledgements or confirmed delivery notifications or both. Using "strong RegIDs" prevents the source from sending messages until the store's source-activity-timeout expires on the store. After the activity timeout expires, the source can reregister.
7014, 6840	Java applications that create multiple receivers for the same topic in the same context may shut down unexpectedly when calling <code>Dispose()</code> if the topic is in a persistent data stream. Informatica recommends that you either avoid concurrent calls to <code>Dispose()</code> or use a single receiver for each topic and use application code to deliver messages to different callbacks.
7937, 7771	With liveness detection enabled for a UMP source and receiver; the receiver sends liveness keepalives to the source regardless of whether or not the receiver successfully registered with the Store.
8728, 8151	You cannot use receiver liveness detection for sources running on a SPARC-based system.
8755	For message consumption acknowledgements, you must not enable both acknowledgement batching and explicit acknowledgements simultaneously. However, if you do enable both, Ultra Messaging does not inhibit this configuration, and you see unexpected behavior.
8870, 8768	When using UMP with Windows, if you terminate a <code>umestored</code> process that was started from a command line window, it might corrupt data that is being written when the process is terminated. This does not apply to <code>umestoreds</code> , which runs as a service. To avoid this problem, ensure that sources are quiescent when stopping the Windows <code>umestored</code> process.

3.3 Queuing Known Limitations

The following open limitations fixes apply to the UMQ product.

Change Request	Description
8596, 8525	<p>If you attempt to disable ULB message reassignment by setting option umq_ulb_application_set_message_reassignment_timeout (source) to 'x: 0', and option umq_ulb_application_set_message_lifetime (source) is already set to the default value of 0, in-flight messages are not discarding.</p> <p>As a workaround, to inhibit message reassignment use option umq_ulb_application_set_message_max_reassignments (source) set to 1, and leave umq_ulb_application_set_message_reassignment_timeout (source) at its default value.</p>
9211	Option umq_queue_activity_timeout (context) has no effect.
9473, 9444	<p>The default paging or "cursor" settings for ActiveMQ might cause poor performance when multiple JMS durable subscribers are added to the same destination. This might cause logs messages similar to the following display:</p> <pre>2014-11-05 17:19:17,205 WARN duplicate message from store ID:Server-45448-1415232622173-1:1:1:1:49, redirecting for dlq processing org.apache.activemq.broker.region.Topic ActiveMQ Transport: tcp:///10.0.0.1:45449@21868</pre> <p>You can improve performance by changing the cursor used by durable subscribers to a vm cursor. However, you then lose the ability in the broker to page messages from memory to disk as memory usage increases. To change to vm cursors for subscribers, update the activemq.xml config file as shown in the following example:</p> <pre><policyEntry topic="\>" > ... <pendingSubscriberPolicy> <vmCursor /> </pendingSubscriberPolicy> <pendingDurableSubscriberPolicy> <vmDurableCursor/> </pendingDurableSubscriberPolicy> ... </policyEntry></pre>
9474, 9446	If a broker restarts and is later elected as active, previously assigned messages might be reassigned to consumers without being marked as redelivered.

3.4 Dynamic Router Known Limitations

The following open limitations apply to the Dynamic Routing Option (DRO).

Change Request	Description
9718	The encryption feature does not implement elliptic curve encryption. Cipher suites with "ECDH", "ECDHE", and "ECDSA" cannot be used. Use "DHE" and "RSA" instead.
3548	The UM Router cannot run as a standalone Windows service.
8736, 8082	When shutting down and restarting a UM Router, the UM Router sometimes does not complete its shutdown fully, requiring that it be forcible killed.

Change Request	Description
8858, 8678	<p>When starting a UM Router on Windows with one or more peer portals configured with dual TCP, and the peer is unable to connect to the other peer, The UM Router might log the following message:</p> <p>CoreApi-5688-3799: WSAEventSelect: (10038) An operation was attempted on something that is not a socket.</p> <p>If this happens, to connect to the other peer you must restart the UM Router.</p>
Restriction	You cannot configure two endpoint portals on a UM Router to have the same adjacent domain id.
Restriction	You cannot configure two peer portals on a UM Router to connect to the same adjacent UM Router.
Restriction	The UM Router ignores UM Spectrum configuration options, which results in UM Routers forwarding all channel data without applying any filtering. See Spectrum .

Chapter 4

Upgrade Procedure

Most upgrades from one version to the next are straightforward and uneventful. This is because Informatica strives to introduce new features without affecting existing functionality.

However, users are strongly encouraged to read and understand the release notes for each intermediate version between their existing system and their intended upgrade target.

In particular, there are four areas that deserve special consideration:

- Default values for configuration options.
- Deprecations.
- Upgrading brokered queuing and JMS applications to 6.8 or beyond from pre-6.8.
- Upgrading to 6.0 or beyond from pre-6.0 versions.

Note

Important: If you are upgrading from Ultra Messaging Version 5.x or earlier, you must recompile and relink your applications to this version's libraries.

4.1 Default Values for Configuration Options

While Informatica attempts to set configuration option defaults to values that represent good trade offs, there are times when new information or customer feedback demonstrates that a different value delivers more value for the average customer. In many cases, customers will benefit from these new values. However, it is also possible that for specific use cases, a previous default value is better.

Informatica recommends that users evaluate each changed default individually and determine if the new value is appropriate for their use cases. In those cases where the previous default is better, the user must explicitly set the option to that value.

Contact Informatica Support for more information and deeper understanding of UM's configuration options.

4.2 Deprecations

To make UM more maintainable, functionality is sometimes removed from UM. Informatica understands that this can be disruptive to users, so we try to restrict such deprecations to features that, to our knowledge, are not needed

by our users.

Informatica recommends careful reading of each version's release notes to learn which features are being deprecated. If you discover that a feature is deprecated which is important to your use case, please contact Informatica Support.

4.3 Upgrading Brokered Queuing and JMS

In UM version 6.8, the brokered queuing functionality and JMS API was re-implemented with an enhanced ActiveMQ-based JMS/AMQP broker. In many cases, the UM queuing API can still be used with few if any changes.

Note

Important: If you are upgrading non-ULB queuing applications from versions prior to 6.8, note that this upgrade has a major impact on your applications and deployment. The umestored daemon no longer implements queuing functionality; instead, the UMQ client API now uses the AMQP protocol to communicate with an ActiveMQ broker for queuing functionality. You must rebuild applications, redeploy daemons, and perform a flash cut over. Also, review your applications for functionalities that are deprecated in UMQ 6.8 or later. UMQ clients using version 6.8 or later will not interoperate with a pre-6.8 umestored-based queue broker. UMQ clients using versions earlier than 6.8 will not interoperate with an ActiveMQ broker. For more information, see the Guide for Queuing.

See **Queuing Upgrade Considerations** for more information about upgrading brokered queuing applications.

4.4 Upgrading From Version Pre-6.0

With the introduction of the UM Router, several changes needed to be made to the "wire format" of UM command and control messages. This resulted in 6.0-based applications not being interoperable with pre-6.0 applications. For many customers, this did not pose a significant problem since they were able to upgrade all of their systems in a single step. I.e. there was no need for pre-6.0 applications to exchange messages with 6.0 applications.

However, for many customers, this kind of "flash" upgrade is simply not practical. Therefore, starting with UM version 6.7, an upgrade path was developed which allows different applications to be upgraded at different times, while still allowing those mixed-version applications to exchange messages.

Note

When you upgrade from a previous version, you must set option **ume_store_activity_timeout (source)** to 10,000 and UMP store option **keepalive-interval** to 3,000. This avoids unexpected behavior during a phased upgrade.

To upgrade Ultra Messaging from version 5.3.x to version 6.7 and beyond, perform the following procedure:

1. If the current installation is earlier than 5.3, upgrade all applications and components except UMDS servers to version 5.3.6 or later.
 2. Pre-6.0 versions of UM are not ABI compatible with 6.0 and beyond. C language application programs must be re-compiled.
 3. If using UMP stores, ensure that all source settings for option **ume_store_behavior (source)** are set to 'qc' or 'quorum-consensus', and not 'rr' or 'round-robin'.
 4. Upgrade all lbmrd resolver daemons to this release. No configuration changes are required.
-

5. If using UMP stores, upgrade all umestored daemons in groupings that observe the following requirements:

- A single source cannot specify a mixture of 5.3.x and this release's umestored daemons.
- Separate sources on the same context can specify daemons of different versions.
- Do not start upgraded daemons with 5.3.x state and cache files.

Note

When upgrading from 5.3.x, if you neglect to set **compatibility_include_pre_um_6_0_behavior (context)** to 1, receivers incorrectly receive messages without persistence.

6. Upgrade source and receiver applications. You can upgrade these one by one, in any order. As you upgrade each application, set the following configuration options:

- **compatibility_include_pre_um_6_0_behavior (context)** to 1
- **transport_tcp_use_session_id (source)** to 0

These settings must remain during the period when the network contains both applications based on 5.3.x and this release.

7. If you use Late Join or OTR, review the settings for Late Join or OTR per-message timeout options. Note that option **otr_request_duration (receiver)** was deprecated in 6.0, and the following options are added:

- **late_join_info_request_interval (receiver)**
- **late_join_info_request_maximum (receiver)**
- **retransmit_request_message_timeout (receiver)**
- **otr_message_caching_threshold (receiver)**
- **otr_request_message_timeout (receiver)**

8. After all components are upgraded to this release, set the following configuration options:

- **compatibility_include_pre_um_6_0_behavior (context)** to 0
- **transport_tcp_use_session_id (source)** to 1

4.4.1 Log Messages During Upgrade

For 5.3.x applications on the same TRD as this release, you might see the following log messages:

LBMC unknown next header 64

Informational. You can ignore this message. You can reduce the frequency of this message by increasing the value for context option **response_tcp_deletion_timeout (context)**.

LBMR Topic Info Record malformed. Dropping remainder

Issued for every TIR that contains a TCP Session ID. You must set option **transport_tcp_use_session_id (source)** to 0 during the interim migration period.

LOG Level 5: Core-5688-3387: LBMR Extended Type 0x7 incorrect (10.29.3.49:40633 len 102). [...D.....<.....1!.....(>.....

Informational. This occurs if UMP store umestored daemons are present and configured with a discoverable name. For each name advertisement, all contexts drop the packet and log this warning.

[WARNING]: Core-5688-516: LBMC cntl unknown next header: 55.

Informational. Issued once per transport session when receiving an SRI from a UMP source.

Chapter 5

UM Version 6.11.1

The most-significant update to UM version 6.11.1 is a set of performance improvements to the Persistent Store daemon. Some performance improvements were also applied to the data receive path for applications.

Attention

See [Deprecations for 6.11.1](#) for important information regarding future platform support for UM daemons.

5.1 Enhancements for 6.11.1

5.1.1 Streaming Enhancements for 6.11.1

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- Performance improvements made to the data reception code paths, including reduction in calls to `gettimeofday()`, and some localized code optimizations.

5.1.2 Persistence Enhancements for 6.11.1

The following new features and enhancements apply to UMP and UMQ products.

- The Persistent Store benefits from the performance enhancements made on the data receive path. (See also [Persistence Fixed Limitations for 6.11.1.](#))

5.1.3 Queuing Enhancements for 6.11.1

The following new features and enhancements apply to the UMQ product.

- None.

5.1.4 Dynamic Router Enhancements for 6.11.1

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- None.

5.2 Fixed Limitations for 6.11.1

5.2.1 Streaming Fixed Limitations for 6.11.1

The following bug fixes apply to UMS, UMP, and UMQ products.

10385, 10280	Enhancement: <i>Reduced number of gettimeofday() calls in context loop. This lowers receive-side latency and CPU load.</i>
--------------	--

5.2.2 Persistence Fixed Limitations for 6.11.1

The following bug fixes apply to UMP and UMQ products.

10332, 10409	FIXED: <i>When the Persistent Store is configured to publish Daemon Statistics, there is a small chance that it will fail to initialize properly. It might log the error:</i> [CRITICAL]: Store-10184-80: umestore_dmon_publish: Core↵ Api-5688-2996: src must be valid <i>and could exit with a segmentation fault.</i>
10400, 10401	FIXED: <i>Bug in Persistent Store leads to performance degradation over time. (This bug has existed in the Store for many years.)</i>
10404, 10411	FIXED: <i>When using RPP with repository-type "disk", it is possible for a blocking receiver to get unrecoverable loss. This loss can be accompanied by the following warning in the Persistent Store log file:</i> [WARNING]: WARNING: failed assertion [ntohl(ohdr->sqn)==rec->sqn]
10406, 10407	FIXED: <i>A bug fix added to 6.10.1 accidentally introduced a significant performance reduction in the UMP Persistent Store. This regression is repaired in 6.11.1.</i>

5.2.3 Queuing Fixed Limitations for 6.11.1

The following bug fixes apply to the UMQ product.

- None.

5.2.4 Dynamic Router Fixed Limitations for 6.11.1

The following bug fixes apply to the Dynamic Routing Option (DRO).

- None.

5.3 Deprecations for 6.11.1

Starting with UM version 6.12, the UM daemons will no longer be available on certain platforms.

Note

User applications will continue to be fully supported on all platforms. These deprecations only apply to UM daemon executables, such as the Store Daemon and the Dynamic Router daemon.

UM daemons will continue to be available on the following platforms:

- Persistent Store daemon ("stored"): Linux 64-bit, Windows 64-bit.
- UM Dynamic Router daemon ("tnwgd"): Linux 64-bit, Windows 64 and 32-bit.
- UM Manager daemon ("ummd"): Linux 64-bit, Windows 64-bit.
- UM Resolution daemon ("lbmrdr"): Linux 64-bit, Windows 64-bit.
- UM Queue Broker (ActiveMQ): Linux 64-bit, Windows 64-bit.

The planned deprecated daemon platforms are:

- Solaris (X86 and Sparc)
- AIX
- MacOS
- Linux 32-bit
- Windows 32-bit (except Dynamic Router "tnwgd", which will continue on 32-bit)

User applications deployed on these platforms will fully interoperate with daemons running on the reduced set of platforms.

Please contact support if any of these planned deprecations interfere with your existing or planned deployments.

Chapter 6

UM Version 6.11

6.1 Enhancements for 6.11

6.1.1 Streaming Enhancements for 6.11

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- **Transport Services Provider (XSP).** XSP is a new method of assigning receiver transport sessions to independent threads. It replaces the earlier limited **Multi-Transport Threads** feature. XSP gives the application control over how received transport sessions are assigned to threads. For more information, see **Transport Services Provider (XSP)**.
- **Enhanced Smart Source feature.** The **Smart Sources** feature now supports more functionality. Some of this support is limited, so please check the corresponding sections for further information.
 - **Smart Sources and Persistence** is now supported,
 - LBT-RU transport type is now supported,
 - **Smart Sources and Spectrum** is now supported,
 - **Smart Sources and Memory Management** is now supported,
 - **Smart Sources and Message Properties** is now supported.

For this UM version, the Smart Source API is not supported in .net.

- **Rolling Log Files for Persistent Store and UM Router (DRO).** To prevent unbounded disk file growth, the Persistent Store and the UM Router now support rolling log files. See **Store Rolling Logs** and **UM Router Rolling Logs** for details.

6.1.2 Persistence Enhancements for 6.11

The following new features and enhancements apply to UMP and UMQ products.

- Smart Sources now support Persistence. See **Smart Sources and Persistence**. (See also [Streaming Enhancements for 6.11](#) for more enhancements to Smart Sources.)
- **Store Daemon Statistics.** A new method of monitoring the Persistent Store daemon. The same information which is currently available on the daemons' web monitors can now be published on a normal UM topic. See **Daemon Statistics**.

6.1.3 Queuing Enhancements for 6.11

The following new features and enhancements apply to the UMQ product.

- None.

6.1.4 Dynamic Router Enhancements for 6.11

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- **Dynamic Router Daemon Statistics.** A new method of monitoring the UM Dynamic Router daemon. The same information which is currently available on the daemons' web monitors can now be published on a normal UM topic. See **Daemon Statistics**.
- The UM Router now logs an informational message each time its internal routing table changes giving the reason for the change. Messages of this form will be printed during normal startup, to give status as the UM Router network establishes connectivity, and they will also be printed when a disruption causes a change in connectivity. The messages are of the form "Gwd-9574-xx".

6.2 Fixed Limitations for 6.11

6.2.1 Streaming Fixed Limitations for 6.11

The following bug fixes apply to UMS, UMP, and UMQ products.

10260	FIXED: <i>When using Smart Source, a source processing NAKs can fatal assert: "(bvec->flags & ~LBTRM_BVEC_SMART_SOURCE_FLAG) == 0".</i>
10178	FIXED: <i>An unnecessary sprintf() initializing a string buffer is being called from <code>lbm_context_↵ process_events()</code>. The sprintf() was moved to context creation.</i>
10135	FIXED: <i>When using the JAVA interface, if the source has <code>ume_message_stability_↵ notification (source)</code> set to <code>LBM_SRC_TOPIC_ATTR_UME_STABLE_EVENT_PER_MES_↵ SAGE</code>, you could get an <code>ArrayIndexOutOfBoundsException</code> on the source event callback.</i>

10087	FIXED: Applications and daemons are at risk of a bus error on architectures that require alignment (SPARC, Power). These happen very rarely, and are associated with UIM messages in a UM Routed (DRO) environment. (Note: UIMs are involved with many features, such as Late Join, Persistence, and ULB.)
10085	FIXED: Interoperation between little-endian and big-endian architectures was broken in UM version 6.10. This problem also interferes with interoperation between 6.10 and any other version of UM, on big-endian architectures. This affects systems that use any combination of UM Routers (DRO), Persistence, or Hot Failover. Users with big-endian architectures are advised to avoid UM version 6.10.
10079	FIXED: Java and .NET programs that use the topic resolution callback by overriding the context object's onResolverEvent() method can generate a segmentation fault, resulting in a JVM dump.
10071, 10072	FIXED: SDM: cloning an attribute object does not copy over the state of the "name_tree" attribute.
10048	FIXED: If a Java application improperly calls respond() for a message after it had been disposed, a crash results. It is requested that respond() instead throw an exception, to make it easier for application bugs to be diagnosed.
9973	FIXED: If lbm_send_request() is called on a heavily-loaded system, and a response is returned very quickly, the "lbm_request_t **reqp" parameter can be returned with NULL.
9963, 10172	FIXED: In UM version 6.7 a change was made which had the unintended consequence of unnecessarily increasing network traffic related to "final advertisements" (see resolver_send_final_advertisements (source)).
9913	Source Send call from within context callbacks is inconsistently documented. The C, Java, and .NET API documents have been updated to be consistent and correct related to context thread callbacks. See lbm_src_send(), lbm_src_send_ex(), lbm_src_sendv(), lbm_src_sendv_ex(), lbm_send_request(), lbm_send_request_ex(), lbm_hf_src_send(), lbm_hf_src_send_ex(), lbm_hf_src_send_rcv_reset(), lbm_hf_src_sendv(), lbm_hf_src_sendv_ex(), lbm_send_response().
9829	FIXED: If a Java application improperly attempts to close a context object before all child objects (sources, receivers) are closed, a crash can result. Now, if an application attempts to close a context that still contains active sources or receivers, a warning is logged and the context is not closed. However, no exception is thrown from the context close; it returns normally. The user must use the error log to identify the application bug.
9816	FIXED: The documentation did not inform the user that it is necessary to have a consistent value for transport_lbtrm_datagram_max_size (context). A note has been added to resolver_datagram_max_size (context), transport_lbtsmx_datagram_max_size (source), transport_lbtrm_datagram_max_size (context), transport_lbtru_datagram_max_size (context), transport_tcp_datagram_max_size (context), transport_lbtipc_datagram_max_size (context), transport_lbtrdma_datagram_max_size (context).
9606	FIXED: Java properties don't support redundant keys. This creates a problem for the resolver_unicast_daemon (context) configuration option which needs to be specified multiple times for redundant lbmrd operation. This does not work with Java Properties. The solution was to modify resolver_unicast_daemon (context) to allow multiple servers to be specified in a single specification, separated by commas or semicolons. For example: cattr.setProperty("resolver_unicast_daemon", "10.3.29.181:5555,10.3.29.185:5555,10.3.29.190:5555");

5555	<p>FIXED: <i>Message Properties are incompatible with the Spectrum feature. Both cannot be used for the same message.</i></p> <p>Now a message can be sent both with message properties and to a Spectrum channel.</p>
7763	<p>This bug cannot be reproduced in any recent version of UM. The bug is assumed to be FIXED: <i>You cannot send messages larger than 65,535 bytes over the LBT-IPC transport when you set option ordered_delivery (receiver) to 0 (zero) unless you also set transport_lbtipc_behavior (source) to 'receiver_paced'.</i></p>

6.2.2 Persistence Fixed Limitations for 6.11

The following bug fixes apply to UMP and UMQ products.

10231	<p>FIXED: <i>The Persistent Store Web Monitor displays truncated values for "Retransmission Request Received Rate" and "Retransmission Request Service Rate".</i></p> <p>The decimal portion is now correctly displayed.</p>
6088, 8660, 8955, 4899	<p>FIXED: <i>Persistent Store log files grow without bounds.</i></p> <p>Rolling log files are now implemented in the Store and in the UM Router. See Store Rolling Logs and UM Router Rolling Logs for details.</p>
10201	<p>FIXED: <i>In a scenario where messages are getting lost and proactive retransmissions are being used between the source and the store, it is possible to get into a state where the source would be infinitely stuck on flight size.</i></p>
10196	<p>FIXED: <i>When a store is not keeping up with incoming messages to the point where it has to intentionally drop incoming messages, there is a possible scenario where the source can get stuck, blocked on flight size.</i></p>
10183, 6324	<p>FIXED: <i>When the store is overwhelmed with incoming messages and has to intentionally drop some messages, it is possible for the state associated with the store to get corrupted and, depending on the next sequence of events, one or more of the following messages can be logged:</i></p> <ul style="list-style-type: none"> • <i>FATAL: failed assertion [buff->len<=LBMUIM_APDU]</i> • <i>WARNING: failed assertion [ntohl(ohdr->sqn)==rec->sqn]</i> • <i>WARNING: failed assertion [ntohs(ohdr->msglen)<=rxbuff->maxlen]</i> • <i>WARNING: failed assertion [ntohs(ohdr->msglen)==rec->disk←_len]</i> <p><i>followed by either an abort or a segmentation fault.</i></p> <p>Checks have been added so that the store's state won't be corrupted.</p>
10176	<p>The description of message Store-5688-5574 has been corrected.</p>
10063, 10064	<p>FIXED: <i>Version 6.10 of the Persistent Store had a condition that could cause rare segmentation faults. When it happens, it is usually associated with the store falling behind the source.</i></p>
10036	<p>FIXED: <i>When using an RPP non-blocking receiver, the source blocks once the store's disk repository fills up.</i></p> <p>Now, when the store's repository fills up, messages will be deleted if only non-blocking acks are outstanding.</p>

6.2.3 Queuing Fixed Limitations for 6.11

The following bug fixes apply to the UMQ product.

- None.

6.2.4 Dynamic Router Fixed Limitations for 6.11

The following bug fixes apply to the Dynamic Routing Option (DRO).

10194	FIXED: <i>When Peer Links attempt to establish connections, there is a window during which a socket could get closed just before attempting to complete the connection. This can result in a fatal assert "handle==conn_mgr->sock->sock", or could simply result in the peer link not making any more connection attempts, leaving the link permanently disconnected.</i>
10188	FIXED: <i>An issue with routing Multicast Immediate Messages through a DRO running SPARC 32-bit systems which caused a bus error.</i>
10163	FIXED: <i>There are several instances of random message corruption which can cause the UM Router to crash.</i> Defensive checks have been added to protect the DRO against malformed messages.
10051	FIXED: <i>Large UIM messages can become corrupted while transiting a DRO network. It happens when message fragments are either dropped (due to link overload) or become out of order.</i>
10050	FIXED: <i>In UM 6.10, the UM Router (DRO) crashes when it is paired with the UMS or UMP version of the UM library.</i>

Chapter 7

UM Version 6.10.1

The most-significant update to UM version 6.10.1 is a set of bug fixes.

7.1 Enhancements for 6.10.1

7.1.1 Streaming Enhancements for 6.10.1

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- None.

7.1.2 Persistence Enhancements for 6.10.1

The following new features and enhancements apply to UMP and UMQ products.

- None.

7.1.3 Queuing Enhancements for 6.10.1

The following new features and enhancements apply to the UMQ product.

- None.

7.1.4 Dynamic Router Enhancements for 6.10.1

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- None.

7.2 Fixed Limitations for 6.10.1

7.2.1 Streaming Fixed Limitations for 6.10.1

The following bug fixes apply to UMS, UMP, and UMQ products.

Change Request	Description
10051, 10130	FIXED: Large UIM messages, such as UM responses, can become corrupted when transferred between Topic Resolution Domains by a UM Router (DRO).
10085, 10086	FIXED: For UM version 6.10 (only) running on "big-endian" machines (e.g. SPARC), 64-bit values in UM packet headers were not properly swapped, with the result that for certain features, big-endian machines cannot interoperate with little-endian machines. The problem is most visible with the ULB feature, and the UM Router (DRO).
10050, 10054	FIXED: For UMM version 6.10 (only), the UM Router (DRO) is not able to run on the UMS or UMP products' "lbn" libraries.
10106	FIXED: Documentation related to several configuration options, including <code>transport_tcp↵_sender_socket_buffer (source)</code> , <code>response_session_sender_socket_buffer (context)</code> , and <code>umecfgrepositorysizethreshold</code> .

7.2.2 Persistence Fixed Limitations for 6.10.1

The following bug fixes apply to UMP and UMQ products.

Change Request	Description
9940, 10131	FIXED: The Proactive Retransmission feature between a persistent source and the Persistent Store did not work correctly when the Receiver-Paced Persistence (RPP) feature was enabled.
10050, 10054	FIXED: For UMM version 6.10 (only), the UM Router (DRO) is not able to run on the UMS or UMP products' "lbn" libraries.

7.2.3 Queuing Fixed Limitations for 6.10.1

The following bug fixes apply to the UMQ product.

- None.

7.2.4 Dynamic Router Fixed Limitations for 6.10.1

The following bug fixes apply to the Dynamic Routing Option (DRO).

Change Request	Description
10050, 10054	FIXED: <i>For UMM version 6.10 (only), the UM Router (DRO) is not able to run on the UMS or UMP products' "lbm" libraries.</i>

Chapter 8

UM Version 6.10

The most-significant updates to UM version 6.10 are related to improving performance (latency, jitter, and throughput).

8.1 Enhancements for 6.10

8.1.1 Streaming Enhancements for 6.10

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- **IPC performance improvement.** Changed the receiver-side dequeuing algorithm to avoid semaphore operations when possible. Also see configuration option **transport_lbtipc_pend_behavior_linger_loop_count (context)**.
- **NCF reduction.** Changed the LBT-RM reliability algorithm to reduce the number of NCF control messages sent during periods of heavy loss. The change reduces the impact of loss on healthy receivers, but might slightly increase the load on the source.
- **Host names in configuration files.** For configuration options that take an IP address, it is now permissible to supply a host name. UM will resolve the name to an IP address when the configuration file is read.
- **Non-contending Statistics retrieval.** Statistics gathering API functions `'lbm*_retrieve*stats()'` no longer contend with the message receive path (context thread). Retrieving statistics no longer introduces latency outliers in normal message flow. Note that a given sample of statistics no longer necessarily represents a single snapshot in time. For example, it is possible that between two samples, the field `'bytes_rcved'` could change but `'lbm_msgs_rcved'` does not. NOTE: brokered queuing contexts are not included in this enhancement.
- **Smart Sources.** A new, somewhat restrictive API for sending messages which eliminates looping, locking, and dynamic memory allocation/freeing, resulting in a significant reduction in send time and elimination of variance in execution time (jitter). Available for C and Java. See **Smart Sources**.
- **Send path performance improvement.** For UDP-based protocols (LBT-RM and LBT-RU), an algorithmic optimization was made to eliminate two memory allocations and frees per send call. This is for established sources (i.e. this optimization is not related to Smart Sources). Reduction of memory allocations and deallocations reduces send time, but more importantly reduces latency outliers (jitter).

- **Send message direct to a source.** Unicast Immediate Messages have always allowed the use of IP address and request port of the destination application. As of version 6.10, a "source string" can also be used as the destination. This allows a receiving application to send a message directly to a newly subscribed source application. See the Concepts Guide section **Sending to Sources**.
- **Default interface configuration option.** This new configuration option simplifies the previously error-prone process of configuring interfaces for different features. If all or most features should use the same interface, it is no longer necessary to configure each interface option. Instead, specify the **default_interface (context)** once and then optionally specify any exceptions. (See below for affected configuration options.)
- **Increased send-side socket buffer for TR.** The send-side socket buffers for Topic Resolution has been increased to 1MB. This will reduce the possibility of send-side loss due to EWOULDBLOCK on socket sends.
- **HTML documentation.** Version 6.10 contains the first phase of a conversion of UM documentation to a different tool chain which produces manuals in 3 forms: multi-page HTML, single-page HTML, and PDF. Manuals converted in 6.10: "Concepts Guide", "Configuration Guide", "Operations Guide", and "Guide to Queuing". Additional manuals will be converted in a future release. In addition, Doxygen is now being used to produce .NET and Java API documentation.
- **Context delay warning.** A UM context keeps track of how long each loop of its context thread takes. As of UM 6.10, it now logs a warning (Core-5688-1882 for embedded mode, Core-5688-1891 for sequential mode) if a loop exceeds 1500 milliseconds (1.5 s). This behavior can be modified by setting the environment variable LBM_CONTEXT_DELAY_WARNING to the number of milliseconds desired, or to zero to disable the warning. Note that if you see this warning, it represents a significant source of latency and should be investigated to determine the root cause.
- **Jar file names changed.** The following shipped Java jar file's names have changed to remove the JDK version number. In the table below, '<vers>' is the UM version number.

Old jar name	New jar name
UMS_<vers>_jdk1.5.0_12.jar	UMS_<vers>.jar
UMSPDM_<vers>_jdk1.5.0_12.jar	UMSPDM_<vers>.jar
UMSSDM_<vers>_jdk1.5.0_12.jar	UMSSDM_<vers>.jar
UMMAPI_1.5.0_12.jar	UMMAPI.jar
UMMD_1.6.0_02.jar	UMMD.jar
UMMGUI_1.6.0_02.jar	UMMGUI.jar

- **The UMSSDM jar file is no longer included in the UMS jar file.** Starting with UM 6.10, Java programs that use SDM need to explicitly include UMSSDM_<vers>.jar in the classpath.
- **Windows .NET API is now included with base Windows package.** It is no longer necessary to download and install .NET separately. The dynamic libraries and example executables are now contained under 'bin\dotnet'.
- **Windows package built with Visual Studio 2012.** The UM packages for Windows are now being built with Visual Studio 2012 (VS 2012). The package is no longer shipped with the Windows Runtime library msvc80.dll. Now UM ships with msvc110.dll. User applications should be built with VS 2012 or beyond.
- **Improved defaults.** The following configuration options have new default values. For systems already in production, please review the options and determine if the improved defaults are appropriate for your application. If not, then you must ensure that the desired values are explicitly configured.

Option	Old default	New default
multiple_receive_maximum_↵ datagrams (context)	1	0
resolver_multicast_interface (context)	INADDR_ANY	Value of default_interface (context)

Option	Old default	New default
resolver_unicast_interface (context)	INADDR_ANY	Value of default_interface (context)
request_tcp_interface (context)	INADDR_ANY	Value of default_interface (context)
response_tcp_interface (context)	INADDR_ANY	Value of default_interface (context)
transport_tcp_interface (source)	INADDR_ANY	Value of default_interface (context)
transport_tcp_interface (receiver)	INADDR_ANY	Value of default_interface (context)
transport_lbtru_interface (source)	INADDR_ANY	Value of default_interface (context)
transport_lbtru_interface (receiver)	INADDR_ANY	Value of default_interface (context)
resolver_multicast_receiver_socket_buffer (context)	0 (OS default)	8388608 (8MB) Increase in socket buffers reduces loss and latency outliers for UDP-based transports with ordered delivery.
resolver_unicast_receiver_socket_buffer (context)	0 (OS default)	8388608 (8MB)
transport_lbtrm_receiver_socket_buffer (context)	524288 (512KB)	8388608 (8MB)
transport_lbtrm_source_socket_buffer (context)	0 (OS default)	1048576 (1MB)
transport_lbtru_receiver_socket_buffer (context)	524288 (512KB)	8388608 (8MB)
transport_lbtru_source_socket_buffer (context)	0 (OS default)	1048576 (1MB)
transport_tcp_sender_socket_buffer (source)	Windows 128000 (125KB)	0 (OS default) The value zero suppresses the setting of the socket buffer, which allows the OS to make use of auto-tuning.
response_session_sender_socket_buffer (context)	Windows 128000 (125KB)	0 (OS default) The value zero suppresses the setting of the socket buffer, which allows the OS to make use of auto-tuning.
transport_lbtrm_rate_interval (context)	100 ms (.1 sec)	10 ms (.01 sec) Reduction of rate interval can improve network stability and prevent loss, but can also increase latency outliers.
otr_request_initial_delay (receiver)	10 sec	2 sec
otr_request_message_timeout (receiver)	20 sec	60 sec

8.1.2 Persistence Enhancements for 6.10

The following new features and enhancements apply to UMP and UMQ products.

- **Improved UM configuration defaults.** The following configuration options have new default values. For systems already in production, please review the options and determine if the improved defaults are appropriate for your application. If not, then you must ensure that the desired values are explicitly configured.

Option	Old default	New default
ume_use_ack_batching (receiver)	0 (no)	1 (yes) Batching of receiver consumption acknowledgements greatly increases system throughput, although it also increases the likelihood of duplicate messages during a recovery operation.
ume_confirmed_delivery_notification (source)	1 (yes)	0 (no) Increases system throughput.
ume_message_stability_timeout (source)	20,000 (20 sec)	5,000 (5 sec) Should be less than NAK generation interval (for UDP-based protocols).

- Improved Store configuration defaults.

Option	Old default	New default
repository-size-threshold	25,165,824 (24MB)	1,024 (1K)

8.1.3 Queuing Enhancements for 6.10

The following new features and enhancements apply to the UMQ product.

- None.

8.1.4 Dynamic Router Enhancements for 6.10

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- Smart-Batch. New method of batching within the UM router which significantly increases throughput without increasing latency (as compared to flushing every message). See <**smart-batch**>.

8.2 Fixed Limitations for 6.10

8.2.1 Streaming Fixed Limitations for 6.10

The following bug fixes apply to UMS, UMP, and UMQ products.

Change Request	Description
10024	FIXED: <i>The API function lbm_transport_source_parse() incorrectly parsed source strings under a wide variety of circumstances.</i>
9824	FIXED: <i>When a .NET program made use of ResolverEventCallback, it could result in a crash with the message, 'A callback was made on a garbage collected delegate...'</i>
9710, 9788	FIXED: <i>The API functions lbm_rcv_topic_attr_dump(), lbm_src_topic_attr_dump(), and lbm_context_topic_attr_dump() generated log messages of the form: 'Core-5688-27 WARNING: ... config variable ... is deprecated. Use ... instead.'</i> <i>One side-effect of this problem is that the Dynamic Router (DRO) logs these messages.</i> <i>This has been corrected since UM version 6.9.1.</i>
9950	FIXED: <i>Corrected certain log messages generated due to Windows socket errors. Previously, the reason for the error was not printed correctly (always said error code was zero). Now it correctly prints the error code returned by Windows.</i>
9877	FIXED: <i>If an LBT-RU source's context thread experiences a significant multi-second delay, due perhaps to scheduling delay on a severely overloaded server, and a receiver is configured to receive multiple datagrams via the multiple_receive_maximum_datagrams (context) option, there is a possibility that the receiver will crash with a fatal assert: '[map != NULL]'.</i>
9939	FIXED: <i>Corrected memory alignment crash when using LZ4 compression on 32-bit SPARC machines.</i>
9981, 9977, 9955	FIXED: <i>UM version 6.9.* have a problem which prevents the optional UM products UMDS, UMCache, and SNMP from working (mul_inet_ntoa).</i>
9722, 9826	FIXED: <i>When the UMDS server component was executed on UMS or UMP library versions prior to 6.9, the server would log: 'CoreApi-5688-22 UMQ not supported'</i> <i>Note that the server would function properly, so the log message could be ignored. In UM 6.10 the log is suppressed.</i>
9957	FIXED: <i>When using Myricom's Datagram Bypass Library (DBL), a fatal assert ('handle==dmux->rcvmsock->sock') and segmentation fault can occur if a receiver is deleted while the receiver is processing incoming datagrams.</i>
9779	Enhancement: Added configuration option transport_lbtru_nak_initial_backoff_interval (receiver) . Setting this greater than zero reduces NAK traffic when UDP datagrams are delivered out of order. However, it also introduces repair latency when a datagram is lost.
4839, 9249, 9950	Enhancement: Updated a number of error log messages to include the proper Windows return code.
9878	Enhancement: In the Java API, if a receiver object is closed more than once, we silently ignore any closes after the first. (Previously an error condition was encountered.)

Change Request	Description
9677	Enhancement: UM Windows package now built with Visual Studio 2012. The Windows runtime library msvcrt110.dll is now included with UM.
9951	Enhancement: print a log message when UM attempts to send a topic resolution message and the send to the socket fails. Most failures are serious and require attention to determine root cause.
9983	Enhancement: include .NET libraries with the base Windows package for all products (UMS, UMP, UMQ). Downloading a separate .NET package is no longer necessary. The DLLs and example program binaries are now under 'bin\dotnet'.
9926	Enhancement: Topic Resolution send-side socket buffers are now set to 1MB. This should prevent "EWOULDBLOCK" errors and the associated loss of TR packets.
9967, 8556, 7793, 2548	Enhancement: reduce the number of NCF control messages sent during periods of heavy loss. The change reduces the impact on healthy receivers, but might slightly increase the load on the source.

8.2.2 Persistence Fixed Limitations for 6.10

The following bug fixes apply to UMP and UMQ products.

Change Request	Description
9953	FIXED: <i>Due to an incorrect print specification, 32-bit applications can occasionally crash with illegal memory access when debug logging is enabled and a persistent session ID is being printed.</i> The print specification has been corrected.
9583	FIXED: <i>When a source requests registration with a store, it is possible for the store's response control message (PREG_RESP) to be lost. The source will re-try the registration, which the store will reject with the log, "RegID xxx is in use by a different source".</i> This has been corrected (a store allows a registered source to re-register).
9920, 9929	FIXED: <i>Under unknown circumstances, a store can exit with inconsistent data in the state and/or cache files. When the store is restarted, this inconsistent data can result in a store crash during initialization.</i> Defensive checks have been added.
9861	FIXED: <i>The UMM tool tip for the source configuration option ume_proactive_keepalive_interval (context) was incorrect. Also, if zero was supplied, CPU utilization jumped to 100%. It is fixed such that zero disables proactive keepalives.</i>
9902, 9925	FIXED: <i>The Store's log file sometimes shows session IDs as negative numbers.</i> These have been corrected to use unsigned.
9458, 9796	FIXED: <i>A minor race condition in the Store which results in one or more warning logs: 'WARNING: failed assertion [repo->disk_holder->disk->offset >= deletesz] at line ...'</i>

Change Request	Description
9938, 9943	Enhancement: If a 5.x persistent source is configured for named stores and is also configured not to query for the store name (via the environment variable 'UME_SKIP_NAME_QUERY'), and a 6.x store becomes non-responsive due to excessive load, and then returns to normal operation, the source will not be able to resolve the store again. This is because a 6.x store cannot be configured to periodically advertise its name indefinitely. In UM version 6.10, the store can be configured to periodically advertise indefinitely via the UM configuration option compatibility_include_pre_um_6_0_behavior (context) . Note that this option is only recommended for use during a limited transition period while customers upgrade from 5.x to 6.x and persistent stores must be configured not to query. When all components are at version 6.x, the use of this configuration option should be discontinued.
9993	Enhancement: Proactive Keepalive feature modified to reduce disk activity due to page faults during periods of low message traffic.
9689	Enhancement: Added Store configuration element <xml-config> to allow the use of XML-based UM configuration files. With this addition, Informatica discourages the use of the environment variable 'LBM_XML_CONFIG_FILENAME'.

8.2.3 Queuing Fixed Limitations for 6.10

The following bug fixes apply to the UMQ product.

- None.

8.2.4 Dynamic Router Fixed Limitations for 6.10

The following bug fixes apply to the Dynamic Routing Option (DRO).

Change Request	Description
9710, 9788	The DRO generated log messages of the form, Core-5688-27 WARNING: ... config variable ... is deprecated. Use ... instead. This was due to a problem in the underlying UM library.
9810	A variety of timing-related issues with single-TCP peer links have been resolved.
9800	If a UM router has a proxy source with queued messages and the proxy source is deleted, the messages get stuck in the queue, which subsequently has less room for future messages. In the worst case, the queue can be stuck in a full state, preventing message forwarding. Proxy sources are deleted as a result of the originating source being deleted, or as a result of all receivers in a remote TRD being deleted, eliminating interest in the topic in that TRD.
9799	If a UM router has a proxy source with queued messages, there is a small timing window during message flow that can cause messages to get stuck in the queue, which subsequently has less room for future messages. In the worst case, the queue can be stuck in a full state, preventing message forwarding. (Similar to 9800, but does not require deletion of proxy source.)

Change Request	Description
9916	If a debug log is being generated, a garbage value is printed in some of the logged messages, making it more difficult for Informatica Support to diagnose problems.

8.3 Deprecations for 6.10

8.3.1 Streaming Deprecations for 6.10

- Adaptive Batching - still operational, but planned to be removed in a future release.
- Explicit Batching - still operational, but planned to be removed in a future release.
- Hypertopics - still operational, but planned to be removed in a future release.
- TCP-LB - still operational, but planned to be removed in a future release.
- RDMA transport - the underlying library is no longer available. This functionality is no longer supported.

8.3.2 Persistence Deprecations for 6.10

- Java APIs deprecated:
 - UMEBlockSrc class
 - LBMContext.setUMQInflight() method
 - LBMContext.setUMQMessageStable() method
- Configuration options deprecated:
 - **retransmit_retention_age_threshold** (source)

8.3.3 Dynamic Router Deprecations for 6.10

- Dual-TCP peer links. UM DRO version 6.10 no longer supports dual-TCP peer links. The XML configuration elements '<tcp>' and '<companion>' will generate errors. Upgrading customers should use **<single-tcp>** instead.
-

Chapter 9

UM Version 6.9.2

The most-significant update to UM version 6.9.1 is the repair of a serious bug in LBT-RM and LBT-RU transports.

9.1 Enhancements for 6.9.2

9.1.1 Streaming Enhancements for 6.9.2

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- None.

9.1.2 Persistence Enhancements for 6.9.2

The following new features and enhancements apply to UMP and UMQ products.

- None.

9.1.3 Queuing Enhancements for 6.9.2

The following new features and enhancements apply to the UMQ product.

- None.

9.1.4 Dynamic Router Enhancements for 6.9.2

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- None.

9.2 Fixed Limitations for 6.9.2

9.2.1 Streaming Fixed Limitations for 6.9.2

The following bug fixes apply to UMS, UMP, and UMQ products.

Change Request	Description
9927	Fixed a problem with UDP transports LBT-RM and LBT-RU where sending a very large number of datagrams (more than 4 billion) results in a failed assertion [objk!=NULL].
9824	Fixed problem in .NET API where using a resolver event callback and/or a source cost callback can cause crashes after garbage collection.

9.2.2 Persistence Fixed Limitations for 6.9.2

The following bug fixes apply to UMP and UMQ products.

Change Request	Description
9902	Fixed problem where session IDs were printed to the Store log file in an inconsistent way, sometimes signed, sometimes unsigned. Session IDs are now always printed unsigned.
9920	Fixed problem in Store where use of Receiver-Paced Persistence can lead to a Store segmentation fault during startup.

9.2.3 Queuing Fixed Limitations for 6.9.2

The following bug fixes apply to the UMQ product.

- None.
-

9.2.4 Dynamic Router Fixed Limitations for 6.9.2

The following bug fixes apply to the Dynamic Routing Option (DRO).

- None.

Chapter 10

UM Version 6.9.1

The most-significant update to UM version 6.9.1 is expansion of the supported platforms:

Linux-X86 (32 and 64 bit), Windows (32 and 64 bit), Solaris-X86 (32 and 64 bit), Solaris-Sparc (32 and 64 bits), Linux-Itanium (64-bit), Linux-Power8 (64-bit, little endian), HP NonStop-X64 (64-bit).

Warning

Due to a serious bug in UM 6.9 and 6.9.1, users of LBT-RM and LBT-RU transports are strongly advised to upgrade to 6.9.2 as soon as possible. The bug causes a crash (fatal assert) after approximately 4 billion datagrams are sent on a transport session.

10.1 Enhancements for 6.9.1

10.1.1 Streaming Enhancements for 6.9.1

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- Addition of Linux-Power8 (64-bit, little endian), HP NonStop-X64 (64-bit) platforms.
- High-resolution Timestamps are now supported for Java applications.

10.1.2 Persistence Enhancements for 6.9.1

The following new features and enhancements apply to UMP and UMQ products.

- None.

10.1.3 Queuing Enhancements for 6.9.1

The following new features and enhancements apply to the UMQ product.

- None.

10.1.4 Dynamic Router Enhancements for 6.9.1

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- None.

10.2 Fixed Limitations for 6.9.1

10.2.1 Streaming Fixed Limitations for 6.9.1

The following bug fixes apply to UMS, UMP, and UMQ products.

Change Request	Description
9729	Fixed a problem where a UDP packet with bad checksum was silently ignored. UDP packets with checksum errors will now generate a warning log message.
9654	Fixed a potential crash with Windows Completions Ports due to memory corruption during a TCP disconnect. This bug was fixed in UM version 6.9, but was not included in the 6.9 release notes due to an oversight.
6102	Fixed a problem where TSNIs were not sent when Source Side Filtering was enabled. TSNIs are now sent.
9722	Enhanced the UMS library to not report certain warnings related to persistence and queuing. These warnings were most apparent with using the UMDS product, which ensures that persistence and queuing behaviors are disabled, but when used with the UMS library, generated warnings. These warnings are not needed and have been removed.

10.2.2 Persistence Fixed Limitations for 6.9.1

The following bug fixes apply to UMP and UMQ products.

Change Request	Description
9707	Fixed a problem with Receiver Paced Persistence where setting the source configuration option ume_write_delay (source) to zero did not override the store's repository-disk-write-delay value. The source can now override the store with ume_write_delay (source) set to zero.
9805	Fixed a problem where certain store log messages associated with registration sometimes caused a segmentation fault on 32-bit Linux.
9481	Fixed a problem where the UMM user interface did not initialize options to the correct default values for the options ume_retransmit_request_outstanding_maximum (receiver) , ume_registration_interval (source) , ume_sri_inter_sri_interval (source) , and ume_sri_max_number_of_sri_per_update (source) . Those options are now initialized to the correct default values.
9462	Enhanced a warning log message related to a store receiving a keepalive message that is not of type "STORE". The message now reports the IP and Port of the sender.
9704	Enhanced a store log message related to source registration errors. The message now reports the IP and Port of the source.
9743	Enhanced the keepalive protocol between the store and a persistent receiver to significantly reduce the keepalive traffic, especially for the case where a persistent receiver exits. This involved the introduction of a new configuration option, ume_proactive_keepalive_interval (context) .

10.2.3 Queuing Fixed Limitations for 6.9.1

The following bug fixes apply to the UMQ product.

10.2.4 Dynamic Router Fixed Limitations for 6.9.1

The following bug fixes apply to the Dynamic Routing Option (DRO).

Change Request	Description
9817	Fixed a topic deafness problem where a temporary surge of messages flowing out of an endpoint portal will put one or more topic queues into a blocked state which does not unblock when the surge is over. Temporarily blocked topic queues are now correctly unblocked.
9755	Fixed a problem where a DRO endpoint could get stuck in an infinite loop sending topic resolution traffic. The problem is associated with the creation and deletion of topics and wildcard receivers which match them. This has been fixed.

Chapter 11

UM Version 6.9

The most-significant update to UM version 6.9 is the introduction of certificate-based security (authentication and encryption).

Warning

Due to a serious bug in UM 6.9 and 6.9.1, users of LBT-RM and LBT-RU transports are strongly advised to upgrade to 6.9.2 as soon as possible. The bug causes a crash (fatal assert) after approximately 4 billion datagrams are sent on a transport session.

11.1 Enhancements for 6.9

11.1.1 Streaming Enhancements for 6.9

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- TCP-based communication can now enable compression and/or encryption. Other protocols (udp, multicast, etc.) are not supported for compression or encryption at this time. See **Encrypted TCP**.
- The LBT-RM transport type now supports a "zero-copy" API for sending messages. See **Zero-Copy Send API**.
- The LBT-RM transport type now supports high-resolution timestamps for packet transmission and reception for customers using Solarflare network interfaces and the Onload driver. See **highresolutiontimestamps**.
- Performance has been improved for sending messages on all transport types by reducing use of dynamic memory. This reduces the send time and also reduces the variance of send times ("jitter").
- Added two new APIs to Java: **com::latencybusters::lbn::LBMessage::getSerializedResponse** and **com::latencybusters::lbn::LBContext::respond**, as well as a new class: **com::latencybusters::lbn::LBSerializedResponse**. These provide a way to receive a UM request, serialize the response object, and then send the response, possibly via a different context.

11.1.2 Persistence Enhancements for 6.9

The following new features and enhancements apply to UMP and UMQ products.

- An API has been added to start and stop a store from inside an application process. See **Callable Store**.
- The store has been enhanced to log thread IDs for each of its internal threads. The log messages are of the form 'Store-8079-x'.
- The Receiver-paced Persistence feature now supports blocking and non-blocking receivers. See **Receiver-paced Persistence Operations**.

11.1.3 Queuing Enhancements for 6.9

The following new features and enhancements apply to the UMQ product.

- None.

11.1.4 Dynamic Router Enhancements for 6.9

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- Peer links may now be compressed and/or encrypted.

11.2 Fixed Limitations for 6.9

11.2.1 Streaming Fixed Limitations for 6.9

The following bug fixes apply to UMS, UMP, and UMQ products.

Change Request	Description
9759	The UM Log Messages chapter of the "Operations Guide" did not correctly order messages numerically. For example, "Core-1234-10" would come before "Core-1234-9". The order has been corrected.
9484, 9566	UM generally reacts to an exhaustion of dynamic memory (heap) with a warning, and attempts to continue execution in a crippled manner. UM has now been changed to react to exhaustion of dynamic memory by triggering a fatal assertion.

Change Request	Description
9582	When the C API lbm_register_fd() is used to register a socket, and an event queue is used to deliver socket events, there is a race condition that can cause delivery of a spurious read event. An application that responds by doing a blocking read can experience an unbounded block. A non-blocking read can return no data. UM now effectively prevents this spurious read event.
9625	When the resolver_unicast_daemon (context) option is read multiple times with the same 'IP:PORT' combination, the context adds multiple references to that lbmrd, leading to wasted resources, potentially exhausting ports in the unicast resolver UDP port range. UM now detects duplicate IP:PORT and maintains a single reference to that lbmrd.
9672	The multiple_receive_maximum_datagrams (context) configuration option enables the use of recvmsg() to improve the efficiency of reading UDP data. However, multiple datagrams were not actually read. No improper behavior results, but the potential efficiency improvement is not delivered. UM now properly enables reception of multiple datagrams.
9591	Under some circumstances, when using the epoll form of FD management, the operating system can return an error to UM that UM does not know how to handle. UM has been triggering a fatal assertion in these cases. Further testing has suggested that some of these unexpected statuses can be benign. So UM has now been changed to use warning asserts when epoll returns an unexpected bad status.
9654	Fixed a potential crash with Windows Completions Ports due to memory corruption during a TCP disconnect.

11.2.2 Persistence Fixed Limitations for 6.9

The following bug fixes apply to UMP and UMQ products.

Change Request	Description
9115	The store now prints BOS and EOS events to its log.
9276	If an RPP store configured for reduced FD storage fails and restarts immediately after a receiver registers, that receiver will sometimes lose messages. Fixed by ensuring that the restarted reduced FD store starts at the correct sequence number.
9626	When the ume_store (source) and/or ume_store_name (source) configuration option is read multiple times referring to the same store, the context adds multiple references to that store, leading to wasted resources. UM now detects duplicate stores and maintains a single reference to that store.

11.2.3 Queuing Fixed Limitations for 6.9

The following bug fixes apply to the UMQ product.

Change Request	Description
9494	UMM did not include the option umq_queue_activity_timeout (context) . It has been added.
9555	When the C API lbn_rcv_topic_attr_setopt() is used to set a value for umq_ulb_↵source_activity_timeout (receiver) , the value was not set correctly. Note that using text-based configuration files worked correctly for that option. The C API function is now corrected.
9640	When the broker configuration option is read multiple times referring to the same broker, the context adds multiple references to that broker, leading to wasted resources. UM now detects duplicate brokers and maintains a single reference to that broker.

11.2.4 Dynamic Router Fixed Limitations for 6.9

The following bug fixes apply to the Dynamic Routing Option (DRO).

Change Request	Description
8880	UM Router daemons using dual-tcp peer links with non-"select" FD management types might issue many 'Core-5688-3808...' log messages. The Router now hard-codes peer links to use select for Unix and wsaeventselect for Windows. This avoids the errors.
9694, 9688	A race condition can cause a portal to become deadlocked and not pass traffic. The portal locking has been corrected.

Chapter 12

UM Version 6.8

The most-significant update to UM version 6.8 is the replacement of the UM brokered queuing and JMS functionality (umestored) with an enhanced Apache ActiveMQ-based JMS/AMQP broker. The existing UMQ queuing API has been adapted to make use of ActiveMQ via the open standard AMQP protocol.

Note that when upgrading a brokered queuing application from a pre-6.8 version, the application source code may need to be changed and re-built.

12.1 Enhancements for 6.8

12.1.1 Streaming Enhancements for 6.8

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- The Multiple Datagram Reception feature can greatly increase receiver performance during message bursts by reading multiple UDP datagrams with one `recvmsg()` system call.

For applications prone to bursts of many messages arriving at a receiver in a very short time, performance can degrade while the receiver reads each message individually. To accelerate message reception during these bursts, enable Multiple Datagram Reception.

This feature applies to all UDP-based sockets, such as topic resolution, MIM, and for transports LBT-RM and LBT-RU. You can use this feature with Linux kernel 2.6.33 or later, and glibc 2.12 or later.

To enable Multiple Datagram Reception, set the configuration option **multiple_receive_maximum_datagrams (context)** to the maximum number of datagrams per socket to receive in one pass.

Note

Due to an implementation defect, this feature does not function correctly in UM version 6.8. Users of version 6.8 are advised to upgrade to UM version 6.9.2 or beyond to take advantage of the performance advantages of this feature.

The following configuration options default values are changed:

Option	Old Default	New Default	Notes
mim_incoming_↔ address (context)	224.10.10.21	0.0.0.0	Disable MIM reception by default to avoid the possibility of default-configured applications sending data that is heard by all contexts.
retransmit_request_↔ outstanding_maximum (receiver)	200 messages	10 messages	Use a more efficient setting for streaming or persistent late join.
response_tcp_↔ deletion_timeout (con- text)	2000 ms	20,000 ms	Use a longer timeout to avoid wasted CPU time caused by unnecessary TCP socket tear down and recreation, along with associated log messages.
resolver_domain_id_↔ active_propagation_↔ timeout (context)	-1	0	This new value where contexts learn of domain IDs from active UM Routers only is now the default.
fd_management_type (context)	wsaeventselect (Win- dows)	wincompport (Windows)	

12.1.2 Persistence Enhancements for 6.8

The following new features and enhancements apply to UMP and UMQ products.

- UM 6.8 introduces the following new API call:
 - C: **lbm_ume_ack_mark_not_outstanding()**
 - Java and .NET: **com::latencybusters::lbm::UMEMessageAck::markNotOutstanding()**

This call updates the internal acking state up to the sequence number provided for throttled recovery, without acking anything.

- You can now configure a Persistent Store to port=0, which selects the actual port number from within the range of the **request_tcp_port_low (context)** and **request_tcp_port_high (context)** configuration options.

The following configuration options default values are changed:

Option	Old Default	New Default	Notes
use_otr (receiver)	0	2 (enables only persistent re- ceivers)	

Option	Old Default	New Default	Notes
ume_registration_interval (source)	500 ms	3000 ms	Reduce overly-aggressive registration activity, which causes excess CPU use. Sources still register very quickly if all stores are available, and only wait for the 'ume_registration_interval' to expire if less than the full list of stores responds to the registration request of the source.
ume_registration_interval (receiver)	500 ms	3000 ms	Reduce overly-aggressive registration activity, which causes excess CPU use. Receivers still register very quickly if all stores are available, and only wait for the 'ume_registration_interval' to expire if less than the full list of stores responds to the registration request of the receiver.
ume_consensus_sequence_number_behavior (source)	majority	highest	For applications that do not reconcile and republish in-flight messages upon store re-registration in the same order they were originally sent, this revised default reduces the chances that a source republishes a previously sent sequence number with a different payload.
ume_sri_inter_sri_interval (source)	100 ms	500 ms	This option, combined with option umesrimaxnumberofsriperupdatesource, determines how long a persistent source advertises stores to receivers after receiving an SRI request. These increased values reduce the likelihood of poor network performance and infrastructure components causing improper UMP registration behavior.
ume_sri_max_number_of_sri_per_update (source)	10	20	

Option	Old Default	New Default	Notes
ume_store_activity_timeout (source)	3000 ms	10,000 ms	This increased value allows for reduced heartbeat traffic, and decreases false declarations of inactive stores. You must set this option and the Persistent Store <code>'keepalive-interval'</code> option together. For information about how these values affect a phased upgrade from earlier versions, see this document's Installation section.
ume_store_behavior (source)	rr	qc	Round-Robin behavior is discontinued.

The following Persistent Store configuration options default values are changed:

Option	Old Default	New Default	Notes
receiver-new-registration-rollback	0	2147483647	The previous default value of 0 resulted in new receivers not requesting any initial state. The new value causes receivers to request all of the data cached in the Persistent Stores that they might be interested in before they become active. You may still need to change these values either with configuration or programmatically.
keepalive-interval	500 ms	3000 ms	This increased value reduces heartbeat traffic, and decreases false declarations of inactive stores. You must set this option and the ume_store↔_activity_timeout (source) option together. For information about how these values affect a phased upgrade from earlier versions, see Upgrading From Version Pre-6.0 .
proxy-election-interval	5,000 ms	60,000 ms	This increased value gives a UMP store more time to create the potentially large number of proxy sources that may result from a large-scale failure event. This reduces the chances of an overload situation in UMP store processes.
source-check-interval	100 ms	750	This increased value reduces CPU use in the UMP store. The previous default was too aggressive and did not significantly improve performance.

12.1.3 Queuing Enhancements for 6.8

The following new features and enhancements apply to the UMQ product.

- Robust commodity-based JMS/AMQP broker features:
 - Fully-managed (pure Java) JMS 1.1 client API
 - AMQP 1.0 wire protocol support for UMQ client API
 - Automatic failover for client to broker connections
 - Inter-operation between UMQ and JMS client APIs

UMQ 6.8 introduces a new architecture for Ultra Messaging Queuing. The `umestored` daemon no longer implements queuing. To support queuing semantics for UMQ clients and both publish/subscribe and queuing semantics for JMS clients, the UMQ client API now supports the Advanced Messaging Queuing Protocol 1.0 wire protocol. Also, the UMQ package now includes Informatica-supported versions of the open-source Apache ActiveMQ broker and ActiveMQ JMS client API.

The Ultra Messaging JMS API and its associated file-based and JNDI configuration methods are now discontinued.

UMQ 6.8 also includes an Ultra Messaging Persistence Adapter for ActiveMQ, which implements a robust, commodity-based replication protocol between brokers using UM Persistence (with instances of `umestored` to persist the replication stream). The UM persistence adapter also implements a quorum-based heartbeat and leader-election protocol to coordinate broker availability.

The Ultra Load Balancing (ULB) feature is unaffected by the changes in this release and continues to operate as before.

- New configuration option: **broker (context)**.
- New Broker transport statistics added:
 - `lbm_rcv_transport_stats_broker_t::msgs_rcved`
 - `lbm_rcv_transport_stats_broker_t::bytes_rcved`
 - `lbm_src_transport_stats_broker_t::msgs_sent`
 - `lbm_src_transport_stats_broker_t::bytes_sent`
- You cannot set option `umq_queue_participation (receiver)` to a value of 2.
- You cannot use the `lbm_rcv_umq_deregister()` function with brokered contexts.
- Sources in a brokered context cannot send Application Headers. Use Message Properties as an alternative.

For more information on upgrading queuing applications, see **Queuing Upgrade Considerations**.

12.1.4 Dynamic Router Enhancements for 6.8

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- None

12.2 Fixed Limitations for 6.8

12.2.1 Streaming Fixed Limitations for 6.8

The following bug fixes apply to UMS, UMP, and UMQ products.

Change Request	Description
2219	A Windows application crashes if you set configuration option fd_management_type (context) to 'select'. The valid option for Windows is 'wsaeventselect'. As a solution, now if you set a Windows application configuration option fd_management_type (context) to 'select', Ultra Messaging returns an error.
6290	<p>In Java, if you reuse the same com::latencybusters::lbm::LBMessageProperties object for every message sent, to adjust those properties dynamically between sends, you are forced to call com::latencybusters::lbm::LBSourceSendExInfo::setMessageProperties.</p> <p>As a solution, the com::latencybusters::lbm::LBSourceSendExInfo object now maintains a reference to an com::latencybusters::lbm::LBMessageProperties object, if it is set, such that the current value of the com::latencybusters::lbm::LBMessageProperties is used at the time of the send, rather than the value of the properties at the time which com::latencybusters::lbm::LBSourceSendExInfo::setMessageProperties() was called.</p> <p>Also, for Java and .NET, the com::latencybusters::lbm::LBSourceSendExInfo::setApplicationHeaderChain(), .setChannelInfo(), .setIndexInfo(), .setMessageProperties(), and .setTotalLifetimeInfo() APIs now update the flags of the com::latencybusters::lbm::LBSourceSendExInfo as appropriate.</p>
8825	An Ultra Messaging Java application might crash when using the com::latencybusters::lbm::UMERegistrationIdExCallbackInfo or UMERecoverySequenceNumberCallbackInfo callback methods.
8926	A hotfix or EBF version of the UM Router installer on Windows fails to install unless the version of Ultra Messaging on the system had exactly the same version number. As a solution, the UM Router installer on Windows now permit installation if the major and minor versions (x.y) are the same.
9015	Message Properties has a memory leak when adding more than 16 message properties.
9024	Ultra Messaging range configuration options pairs do not validate values to ensure that the low range value is less or equal to the high range value.
9025	The lbmpong.c sample application might cause the following fatal assert: FATAL: failed assertion [src->exbq!=NULL] at line 2155 in ../../../../src/lib/lbm/lbmsrc.c
9028	If the function lbm_deserialize_response() has a NULL pointer for the second variable, the function returns an LBM_EINVAL. As a solution, the lbm_deserialize_response() function now fatally asserts if NULL is passed for the 'lbm_context_t*' and/or 'lbm_serialized_response_t*' arguments.
9029	Ultra Messaging Datagram Bypass Layer (DBL) still uses Myricom DBL version 1. As a solution, Ultra Messaging now uses Myricom DBL version 3.

Change Request	Description
9033	A Windows receiver using FD management type wincompport might crash with the following log message: Core-5688-3839: lbm_fd_handle_events line 4463: wincport recv WSA err 0 (Not a WSA error.) from peer not a connected socket
9034	The lbmrd process no longer requires a license to run, but the help menu still references the '-f' command line option/value, which has no effect.
9035	An HFX receiver with option receiver_callback_service_time_enabled (context) option set to 1 might crash or operate without getting any data.
9036	When referenced from an IDE, the Ultra Messaging .NET API libraries do not specify Informatica as creator.
9086	Multi homed windows machines where the IP addresses of at least two NICs have the same number of digits might return the wrong IP address for an adapter name.
9090	For the LBT-RM transport, the 'nak_stm_min' and 'nak_tx_min' statistics might contain incorrect values, such as MAX_INT+1 or 0x100000000, due to being incorrectly initialized. As a solution, these statistics are now correctly initialized to 0.
9092	In a Java sending application, if you set the message properties flag while sending but do not set the actual message properties object, the application might crash.
9077	When you set option resolver_multicast_incoming_address (context) to 0.0.0.0, the Ultra Messaging application exits during topic resolution.
9143	An Ultra Messaging application using fd management type <code>epoll</code> might have file descriptors silently become unregistered.
9146	When delivering a no source notification to a UM wildcard receiver application, it may experience a segmentation fault.
9147	Ultra Messaging unicast applications might process topic resolution data from lbmrd resolver daemons that they are not configured to use.
9152	An Ultra Messaging application using fd management type select might fatally assert with the following message: FATAL: failed assertion [index<fdm->select_fd_count]
9155	The lbm_event_dispatch() function might invoke a user callback before the associated event queue has been shut down. As a solution, the lbm_event_dispatch() function now checks the event queue shutdown status before calling a user callback, but still always processes UNBLOCK events.
9155	When sending serialized Ultra Messaging responses via the UM router, the Ultra Messaging request application might not receive them.
9160	For AIX systems, a setsockopt call for 'IP_ADD_MEMBERSHIP' might fail silently, possibly resulting in an EOS without a BOS. As a solution, if this failure occurs Ultra Messaging now retries the call and generates the following log message: Core-9160-1: INFO: Attempted IP_ADD_MEMBERSHIP on multicast receive socket returned ENOBUFS, retrying.
9169	An application might freeze when a new thread tries to dispatch events after the event queue is shut down.

Change Request	Description
9176	In certain configurations, contexts do not learn Domain IDs as expected. To avoid this, all contexts in the same Topic Resolution Domain (TRD) must use the same value for the option resolver_domain_id_active_propagation_timeout (context) .
9191	In Java, when you send message properties set in an com::latencybusters::lbm::LBMSourceSendExInfo object, garbage collection might corrupt the message. As a solution, com::latencybusters::lbm::LBMSourceSendExInfo objects now maintain a reference to com::latencybusters::lbm::LBMMMessageProperties objects, if set.
9205	In Java, Ultra Messaging ignores the com::latencybusters::lbm::LBMAApplicationHeaderChain object that is passed into the com::latencybusters::lbm::LBMSourceSendExInfo constructor.
9298	If an Ultra Messaging application's context thread processing is delayed significantly, the LBT-RM or LBT-RU receiver activity might time out before the application adds a receiver socket file descriptor to the Ultra Messaging file descriptor set. This causes a memory corruption.
9370	In a Windows application using transport TCP, the application might fatally assert with the following message: FATAL: failed assertion [handle==conn->sock->sock] at line 980 in lbm\lbmtcp.c
9372	When you enable 'LBM_CONTEXT_DELAY_WARNING_THRESHOLD', the context delay value displays a misleading value in seconds. As a solution, the context delay time printed now shows the correct value.
9436	Message selectors that consist of only white space cause a message selector parsing error.

12.2.2 Persistence Fixed Limitations for 6.8

The following bug fixes apply to UMP and UMQ products.

Change Request	Description
4727	If you restart all UMP stores simultaneously, UMP stores cannot recreate proxy sources. As a solution, the UMP Store now maintains information about proxy sources as part of the per-source state, and can elect proxy sources upon restart of the store daemon if necessary. Once restarted, a store waits for the configured source-activity-timeout before it attempts to create a proxy source.
8814	Some UMP store log messages are missing newlines.
9023	A receiver might disable the option to use Late Join without disabling UMP store participation, which is an invalid configuration. As a solution, if a receiver subscribes to a persistent source with ume_use_store (receiver) enabled but use_late_join (receiver) disabled, the use_late_join (receiver) option is automatically enabled for that source only. and you see the following log message: LOG Level 7: Core-6959-1: INFO: Receiver on topic "xyz" has its use_late_join option disabled but has a persistent source on transport TCP:10.29.3.42:14371↵:7d6a8a86[4208529623]. To opt out of Late Join when subscribing to persistent sources, you must also disable the ume_use_store (receiver) option. This will not cause a receiver with use_late_join (receiver) disabled to participate in Late Join from a non-persistent source.

Change Request	Description
9037	If a store with receiver paced persistence enabled experiences transport-level loss, then a UMP source might fail to decrement the flight size count and block indefinitely without sending.
9091	If you run the Java umesrc sample application without specifying a store, the application should exit, but does not. As a solution, the Java umesrc now displays an error message and exits if you do not specify a store.
9094	The .NET umesrc sample application crashes when using the <code>-S</code> command line option to input a valid store IP and port.
9095	The .NET umesrc sample application crashes when using the <code>-S</code> command line option to input a valid store IP and port.
9214	With low flight size settings, in-flight messages might experience minor processing delays.
9302	A store service starting up on Windows might lose data or become unstable if it attempts to read locked cache or state files. As a solution, the store now exits at startup if any cache or state files are found to be locked by another process.
9323	Ultra Messaging might incorrectly drop a message that follows a fragmented message.
9443	<p>A receiver might get late join messages even if use_late_join (receiver) is disabled. This was caused by the original solution to bug 9027, which forcibly enabled use_late_join (receiver) for a receiver that joined a persistent stream from at least one source. This caused the receiver to also Late Join from non-persistent sources on the same topic discovered after the persistent source.</p> <p>As a solution, Late Join participation is now forcibly enabled only with respect to persistent sources; a receiver does not Late Join from non-persistent sources on the same topic if the use_late_join (receiver) option is disabled. See description for bug 9027 for more details.</p>

12.2.3 Queuing Fixed Limitations for 6.8

The following bug fixes apply to the UMQ product.

Change Request	Description
8621	Configurations with multiple slave queue instances can cause data loss, crashes or an inability to restart without removal of cache and state files (resulting in data loss). Informatica recommends deploying only configurations with a single queue instance without slaves. To facilitate failover, configure the <code>'sync-log-filename'</code> , <code>sync-data-filename</code> , and <code>'sync-queue-swap-filename'</code> to write to a shared file system, and use external process management (automatic or manual) to start up a backup queue instance referencing the same files if and only if the active instance fails (i.e. only allow one queue instance to access the files at any time). Note that host failures (e.g., OS, disk, server) may still result in data loss. With this configuration, sync files grow over time, so perform clean restarts (i.e., shut down, delete all files and restart) periodically. Using a shared file system may also impact performance; Informatica recommends holistic system performance characterization prior to production deployment.
8798	You cannot send messages with a zero-length payload.

Change Request	Description
9148	When a Java source uses ULB and sends messages with com::latencybusters::lbm::LBMSourceSendExInfo without a per-send client object, a memory leak occurs. As a solution, the call to the <code>lbm_src_send_count_expected_terminals()</code> function occurs only when there is a per-send client present.
9305	The sample Java application <code>umqrcv.java</code> might produce garbled output.
9318	The .NET sample application <code>umqrcv.cs</code> no longer deregisters upon exit by default. Use the <code>-D</code> command line option.

12.2.4 Dynamic Router Fixed Limitations for 6.8

The following bug fixes apply to the Dynamic Routing Option (DRO).

Change Request	Description
8851	Setting option <code><max-datagram></code> to a value other than the default of 65535 can cause the UM Router to hang. As a solution, the UM Router now discards wrong-sized messages and continues.
8979	Incorrect UM Router endpoint Domain IDs propagate continuously, without a mechanism for resetting other than restarting all contexts in the TRD.
9026	When a UM Router peer link closes the connection, the other peer issues the following log message: <code>Gwd-6033-618: peer portal [GW_0_to_GW_2] inbound connection destroyed due to socket failure</code> As a solution the other peer now issues the following correct log message: <code>Gwd-6033-618: peer portal [name] inbound connection destroyed due to disconnect</code>
9144, 9153	On UNIX platforms, UM Router daemons using single-tcp peer links might incorrectly declare peer links as inactive, and attempt to tear down and re-establish the peer connection. As a solution, for UNIX platforms the UM Router now forces the peer portal option fd_management_type (context) to 'select' regardless of the user's configuration settings.
9178	In certain configurations using varying values of configuration option resolver_domain_id_active_propagation_timeout (context) , Ultra Messaging might assign incorrect domain ID values.
9299	When using UM Router peer portals, a peer disconnect might result in a deadlock, which causes the UM Router to stop functioning and requires a restart.

12.3 Deprecations for 6.8

12.3.1 Persistence Deprecations for 6.8

- For Persistent Stores, the '<topic>' option type value 'regex' is deprecated.
- The Round-Robin store failover configuration, which was deprecated in UM 6.7, is now removed from Ultra Messaging 6.8 and is no longer available.

12.3.2 Queuing Deprecations for 6.8

The following configuration options are deprecated and no longer function.

- umq_flight_size (context)
- umq_flight_size_behavior (context)
- umq_msg_total_lifetime (context)
- umq_message_retransmission_interval (context)
- umq_message_stability_notification (context)
- umq_queue_check_interval (context)
- umq_queue_name (source)
- umq_queue_participants_only (source)
- umq_queue_query_interval (context)
- umq_require_queue_authentication (context)
- umq_retention_intergroup_stability_behavior (context)
- umq_retention_intragroup_stability_behavior (context)
- umq_retention_intergroup_stability_behavior (source)
- umq_retention_intragroup_stability_behavior (source)

The following C API functions are deprecated and no longer function:

- **lbm_ctx_umq_get_inflight()**
- **lbm_umq_ctx_msg_stable()**
- **lbm_queue_immediate_message()**

The following Java and .NET API functions are deprecated and no longer function:

- UMEBlockSrc class
- LBMContext.setUMQInflight() method
- LBMContext.setUMQMessageStable() method

12.3.3 Dynamic Router Deprecations for 6.8

- The '<regex-pattern>' configuration option element is deprecated.
-

Chapter 13

UM Version 6.7.3

The most-significant update to UM version 6.7.3 is the repair of a critical bug that primarily affects the UM Router (DRO) when using peer links.

13.1 Enhancements for 6.7.3

13.1.1 Streaming Enhancements for 6.7.3

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- Ultra Messaging 6.7.3 introduces the following new Java class and methods:
 - Class: **com::latencybusters::lbm::LBMSerializedResponse**
 - Method: **com::latencybusters::lbm::LBMessage::getSerializedResponse()**
 - Method: **com::latencybusters::lbm::LBContext::respond()**

These methods allow Java applications to extract a serialized response (a byte array representation of the request originator's address) from a received request message, and allow either the same application or another application (if the byte array representation is passed to the other application, e.g., via a standard message) to use that serialized response to send a response to the request originator.

13.1.2 Persistence Enhancements for 6.7.3

The following new features and enhancements apply to UMP and UMQ products.

- None.

13.1.3 Queuing Enhancements for 6.7.3

Note

The brokered queuing and UM JMS functionalities were replaced in version 6.8, so most of these notes do not apply to more-recent versions of UM. However, ULB functionality is retained.

The following new features and enhancements apply to the UMQ product.

- None.

13.1.4 Dynamic Router Enhancements for 6.7.3

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- None.

13.2 Fixed Limitations for 6.7.3

13.2.1 Streaming Fixed Limitations for 6.7.3

The following bug fixes apply to UMS, UMP, and UMQ products.

Change Request	Description
9508	On Windows, an application might crash when a TCP receiver disconnects from a TCP source.
9515	Multi-homed windows machines where the IP addresses of at least two NICs have the same number of digits might return the wrong IP address for an adapter name.
9516	When using UM Router peer portals, a peer disconnect might result in a deadlock, which causes the UM Router to stop functioning and requires a restart.

13.2.2 Persistence Fixed Limitations for 6.7.3

The following bug fixes apply to UMP and UMQ products.

- None.
-

13.2.3 Queuing Fixed Limitations for 6.7.3

Note

The brokered queuing and UM JMS functionalities were replaced in version 6.8, so most of these notes do not apply to more-recent versions of UM. However, ULB functionality is retained.

The following bug fixes apply to the UMQ product.

- None.

13.2.4 Dynamic Router Fixed Limitations for 6.7.3

The following bug fixes apply to the Dynamic Routing Option (DRO).

- None.
-

Chapter 14

UM Version 6.7.2

Ultra Messaging version 6.7.2 consists only of the UMS product for the Stratus OpenVOS platform.

14.1 Enhancements for 6.7.2

14.1.1 Streaming Enhancements for 6.7.2

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- UMS 6.7.2 is available for the Stratus OpenVOS platform.

Chapter 15

UM Version 6.7.1

The most-significant update to UM version 6.7 is the inclusion of the HP NonStop platform for the UMP product. Note that daemons are not supported on NonStop, only client applications.

15.1 Enhancements for 6.7.1

15.1.1 Streaming Enhancements for 6.7.1

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- You can now set how a context learns the ID of its own Topic Resolution Domain (TRD) with option **resolver↔_domain_id_active_propagation_timeout (context)**. This provides more flexibility in situations where you might need to change the Topic Resolution Domain (TRD) ID values for contexts.
- Log messages 'Core-5688-1882' and 'Core-5688-1891' now display delay times in seconds and milliseconds, instead of seconds only.

15.1.2 Persistence Enhancements for 6.7.1

The following new features and enhancements apply to UMP and UMQ products.

- Ultra Messaging Persistence Edition is now available on the HP NonStop platform. For information on limitations, see the [Documentation Introduction](#).

15.1.3 Queuing Enhancements for 6.7.1

Note

The brokered queuing and UM JMS functionalities were replaced in version 6.8, so most of these notes do not apply to more-recent versions of UM. However, ULB functionality is retained.

The following new features and enhancements apply to the UMQ product.

- None.

15.1.4 Dynamic Router Enhancements for 6.7.1

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- None.

15.2 Fixed Limitations for 6.7.1

15.2.1 Streaming Fixed Limitations for 6.7.1

The following bug fixes apply to UMS, UMP, and UMQ products.

Change Request	Description
3944	The 'lbmpong.c' sample application might cause the following fatal assert: FATAL: failed assertion [src->exbq!=NULL] at line 2155 in ../../../../src/lib/lbm/lbmsrc.c
6186	If the function lbm_deserialize_response() has a NULL pointer for the second variable, the function returns an LBM_EINVAL. As a solution, the lbm_deserialize_response() function now fatally asserts if NULL is passed for the 'lbm_context_t*' and/or 'lbm_serialized_response_t*' arguments.
7526	If an Ultra Messaging application's context thread processing is delayed significantly, the LBT-RM or LBT-RU receiver activity might time out before the application adds a receiver socket file descriptor to the Ultra Messaging file descriptor set. This causes a memory corruption.
7945	When delivering a no source notification to a UM wildcard receiver application, it may experience a segmentation fault.
8428	The lbm_event_dispatch() function might invoke a user callback before the associated event queue has been shut down. As a solution, the lbm_event_dispatch() function now checks the event queue shutdown status before calling a user callback, but still always processes UNBLOCK events.
8517	An Ultra Messaging application using fd management type select might fatally assert with the following message: FATAL: failed assertion [index<fdm->select_fd_count]

Change Request	Description
8750	An HFX receiver with option receiver_callback_service_time_enabled (context) option set to 1 might crash or operate without getting any data.
8788	Ultra Messaging Datagram Bypass Layer (DBL) still uses Myricom DBL version 1. As a solution, Ultra Messaging now uses Myricom DBL version 3.
8812	Ultra Messaging range configuration options pairs do not validate values to ensure that the low range value is less or equal to the high range value.
8856	The lbmrd process no longer requires a license to run, but the help menu still references the '-f' command line option/value, which has no effect.
8874	When referenced from an IDE, the Ultra Messaging .NET API libraries do not specify Informatica as creator.
8879	An Ultra Messaging application using fd management type <code>epoll</code> might have file descriptors silently become unregistered.
8898	A Windows receiver using FD management type <code>wincompport</code> might crash with the following log message: Core-5688-3839: lbm_fd_handle_events line 4463: wincport recv WSA err 0 (Not a WSA error.) from peer not a connected socket
8919	For the LBT-RM transport, the lbm_rcv_transport_stats_lbtrm_t::nak_stm_min and lbm_rcv_transport_stats_lbtrm_t::nak_tx_min statistics might contain incorrect values, such as <code>MAX_INT+1</code> or <code>0x100000000</code> , due to being incorrectly initialized. As a solution, these statistics are now correctly initialized to 0.
9013	Ultra Messaging unicast applications might process topic resolution data from lbmrd resolver daemons that they are not configured to use.
9040	When sending serialized Ultra Messaging responses via the UM router, the Ultra Messaging request application might not receive them.
9077	When you set option resolver_multicast_incoming_address (context) to 0.0.0.0, the Ultra Messaging application exits during topic resolution.

15.2.2 Persistence Fixed Limitations for 6.7.1

The following bug fixes apply to UMP and UMQ products.

Change Request	Description
8933	An Ultra Messaging Java application might crash when using the com::latencybusters::lbm::UMERegistrationIdExCallbackInfo or com::latencybusters::lbm::UMERecoverySequenceNumberCallbackInfo callback methods.
5141	If you run the Java umesrc sample application without specifying a store, the application should exit, but does not. As a solution, the Java umesrc now displays an error message and exits if you do not specify a store.
6959	When a UMP receiver subscribes to a UMP source with option ume_use_store (receiver) enabled, the application can not disable the option use_late_join (receiver) .

Change Request	Description
8140	When option source_includes_topic_index (context) is enabled, the source string delivered to the UME recovery sequence number callback does not include the topic index.
8963	If a store with receiver paced persistence enabled experiences transport-level loss, then a UMP source might fail to decrement the flight size count and block indefinitely without sending.
8995	The .NET umesrc sample application crashes when using the <code>-S</code> command line option to input a valid store IP and port.
9010	The .NET umesrc sample application sometimes crashes when the store name length equals 0.

15.2.3 Queuing Fixed Limitations for 6.7.1

Note

The brokered queuing and UM JMS functionalities were replaced in version 6.8, so most of these notes do not apply to more-recent versions of UM. However, ULB functionality is retained.

The following bug fixes apply to the UMQ product.

Change Request	Description
8950	When a Java source uses ULB and sends messages with com::latencybusters::lbm::LBMSourceSendExInfo without a per-send client object, a memory leak occurs. As a solution, the call to the <code>lbm_src_send_count_expected_terminals()</code> function occurs only when there is a per-send client present.
8966	When a queue daemon terminates and restarts with a reset state, the source might resubmit a message that the queue daemon has already stabilized.
8967	When the queue daemon receives messages via UMQ pro-active retransmission, some of these messages might not have the LBM_MSG_FLAG_UMQ_RESUBMITTED flag.

15.2.4 Dynamic Router Fixed Limitations for 6.7.1

The following bug fixes apply to the Dynamic Routing Option (DRO).

Change Request	Description
8859, 8871, 9002	On UNIX platforms, UM Router daemons using single-tcp peer links might incorrectly declare peer links as inactive, and attempt to tear down and re-establish the peer connection. As a solution, for UNIX platforms the UM Router now forces the peer portal option fd_management_type (context) to <code>'select'</code> regardless of the user's configuration settings.

Change Request	Description
8873	<p>When a UM Router peer link closes the connection, the other peer issues the following log message:</p> <pre>Gwd-6033-618: peer portal [GW_0_to_GW_2] inbound connection destroyed due to socket failure</pre> <p>As a solution the other peer now issues the following correct log message:</p> <pre>Gwd-6033-618: peer portal [name] inbound connection destroyed due to disconnect</pre>
8979	<p>Incorrect UM Router endpoint Domain IDs propagate continuously, without a mechanism for resetting other than restarting all contexts in the TRD.</p>

Chapter 16

UM Version 6.7

The most-significant update to UM version 6.7 is interoperability with pre-6.0 versions of UM. Applications and persistent stores at version 6.7 are able to exchange messages and control information with 5.x applications, especially 5.3.6.

However, note that 6.7 applications are not able to interoperate with daemon components of 5.x, including the persistent store and the 5.x gateway. Also, 5.x applications are not able to interoperate with version 6.7 UM Router (DRO).

16.1 Enhancements for 6.7

16.1.1 Streaming Enhancements for 6.7

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- **Receive-Side Batching** - reduces latency and increases throughput in UM applications by bundling multiple messages to deliver. This bypasses the need to perform a boundary crossing for each delivered message. You enable Receiver-Side Batching with configuration option **delivery_control_message_batching (context)**. With C and .NET applications, you must be using an event queue for this feature (this is not a requirement for Java applications).
- The unicast resolver daemon, lbmrd, no longer requires an Ultra Messaging license to run.
- The unicast resolver daemon, lbmrd, now has command-line options to set buffer size in bytes for the receive socket buffer (`-r` flag) and the send socket buffer (`-s` flag).
- In addition to Receive-Side Batching, other improvements have further reduced the latency and increased the throughput of the Java API.
- When an error occurs on a socket while sending an ACK in LBT-RU, UM now generates an EOS immediately.
- You can now set a source activity timeout for LBT-TCP sessions, with new configuration option **transport_tcp_activity_timeout (source)**.
- For UM XML configuration files and UMP store daemon XML configuration files you can use the 'XInclude' mechanism to merge multiple configuration files.

16.1.2 Persistence Enhancements for 6.7

The following new features and enhancements apply to UMP and UMQ products.

- **Throttled Delivery** - you can limit the number of message fragments that a persistent receiver delivers to a receiver application beyond the current highest consumed message fragment. This limits the amount of memory a receiver uses when that receiver consumes messages at rate much slower than the recovery rate from a UMP store or slower than the UMP source stream. To use UMP Throttled Delivery, set the new option **ume_application_outstanding_maximum (receiver)**.
- **Enhanced Quorum/Consensus Store Registration and Message Stability**
 - Quorum/Consensus functionality is enhanced in the following ways:
 - * When setting option **ume_retention_intragroup_stability_behavior (source)** to 'all-active', the source requires, minimally, a quorum of stores to successfully stabilize a message within the group.
 - * The **ume_retention_intragroup_stability_behavior (source)** and **ume_retention_intergroup_stability_behavior (source)** options now apply to registration behavior, as well as message stability behavior. With versions earlier than 6.7, intragroup registration was quorum-based only, and intergroup registration completion always occurred after the first store group was registered.
 - * During the first cycle of the **ume_registration_interval (source)** period, a source does not declare Registration Complete unless all stores in all groups respond. With versions earlier than 6.7, a source declares Registration Complete as soon as all stores in one group respond.
 - * During subsequent cycles of the **ume_registration_interval (source)** period, a source declares Registration Complete as soon as the configured settings for the **ume_retention_intragroup_stability_behavior (source)** and **ume_retention_intergroup_stability_behavior (source)** options are satisfied. With versions earlier than 6.7, a source declares Registration Complete as soon as a quorum of stores in one group respond.
- UM now propagates a source's session ID to receivers. The session ID is visible during registration events as well as during the recovery sequence number callback.

16.1.3 Queuing Enhancements for 6.7

Note

The brokered queuing and UM JMS functionalities were replaced in version 6.8, so most of these notes do not apply to more-recent versions of UM. However, ULB functionality is retained.

The following new features and enhancements apply to the UMQ product.

16.1.4 Dynamic Router Enhancements for 6.7

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- The UM Router now checks configuration option values to ensure that they are within valid ranges.
- For UM Router XML configuration files you can use the 'XInclude' mechanism to merge multiple configuration files.

16.2 Fixed Limitations for 6.7

16.2.1 Streaming Fixed Limitations for 6.7

The following bug fixes apply to UMS, UMP, and UMQ products.

Change Request	Description
2551	With adaptive batching enabled, applications sometimes issue a warning assertion, <code>WARNING: failed assertion [imbq->stats.ttss_mean<=imbq->stats.ttcs_mean]</code> As a solution, since the condition is harmless, the warning is removed.
5242	Web monitor support link text displays, "29West Ultra Messaging Support". This is now changed to display, "Informatica Ultra Messaging Support".
5626	Ultra Messaging client XML configuration validation sometimes fails needlessly. As a solution, XML configuration parsing now tolerates invalid values such that you can use a single XML config file or UMM configuration template across multiple platforms, even if it contains option values that are not valid for all platforms.
7261	If automatic monitoring is misconfigured, instead of silently failing the creation and proceeding with a context or event queue setup, Ultra Messaging now logs the following messages: 'Could not create automon controller: CoreApi-5688-3059: automatic monitoring initialization failed [4099] [Transport module source init function returned -1, <code>lbmaux_context_attr_setopt_from_file()</code> failed CoreApi-5688-4374: invalid configuration line in mcast.cfg, line 1: xxxxxxx
7917	Destroying a source that matches a wildcard pattern, while dispatching an event queue with events from that source, creates a race condition.
7939	LBT-RU does not indicate in the packet when a datagram is a retransmission. As a solution, Ultra Messaging now marks retransmitted LBT-RU datagrams with a retransmit flag. This flag is visible only with packet dissection tools such as Wireshark.
7944	Using the function <code>lbm_context_attr_dup()</code> might cause a <code>src->res_ucast_daemons!=NULL</code> assert when the original attributes do not contain a setting for the UM configuration option resolver_unicast_daemon (context) and when that option is then set after the original is created.
8024	For Late Join, OTR, or Request/Response, Ultra Messaging does not perform NAT translation correctly.
8065	When using UM Routers (DRO), Ultra Messaging applications can cache old data that later can cause errors when creating and deleting wildcard receivers. The Ultra Messaging application context now cleans out this old data when it receives UM Router proxy source delete messages.
8108	Ultra Messaging accepts configuration option resolver_unicast_daemon (context) when it specifies more than one address. As a solution, incorrectly setting this option now issues message <code>CoreApi-7875-1</code> indicating that multiple values are not allowed for this option.
8114	The lbmrd resolver sometimes issues message <code>LOG Level 6: Core-5688-3375: unicast resolver xxx.xxx.xxx.xxx:xxxx went inactive while the resolver is active.</code>

Change Request	Description
8153	When a receiver receives increasingly large messages, a memory leak might occur.
8150	Ultra Messaging may now declare fragments requested for retransmission by a receiver from a store as unrecoverable prior to the retransmission request timeout, if all stores have declared that fragment unrecoverable.
8160	On a 64-bit UNIX platform using a UM XML configuration file, certain context names, event queue names, topic names, and wildcard patterns are not picked up, even though the configuration file specifies them.
8198	Mellanox Ultra Messaging applications with option ud_acceleration (context) set to 1 were forcing the VMA_SELECT_POLL environment variable to 0, resulting in poor latency performance. Applications can now set VMA_SELECT_POLL to the desired value, or can allow the Mellanox default value.
8277	LBT-SMX transport statistics include unsupported request/response statistics. As a solution the field <code>lbm_reqs_rcvd</code> is renamed to 'reserved1' in the lbm_rcv_transport_stats ↵ lbtsmx_t structure.
8283	LBMMON statistics receivers now report correct values for SMX transport statistics.
8295	The lbmrd unicast resolver daemon propagates certain non-LBMR messages to clients. As a solution, lbmrd now ignores these messages.
8444	When using LBT-RU and Source Side Filtering, a <code>bvec->buffs[0] != NULL</code> fatal assert can occur when receivers are connecting, messages are being sent, and there is loss and retransmissions. As a solution, Ultra Messaging now takes the proper lock when sending the Source Side Filtering initialization message.
8557	When you configure LBT-RU context and receiver port ranges to overlap, Ultra Messaging allows this and logs many copies of Core-5688-1796 messages. As a solution, Ultra Messaging now throttles this message as shown in the following example: LOG Level 6: THROTTLED MSG: Core-5688-1796: LBT-RU receiver received unknown packet type 6. Origin: 10.29.↵ 3.187:16001 0.9995 secs. 0 msgs/sec. 0 bps LOG Level 6: ...previous THROTTLED MSG repeated 25 times in 3 seconds
8591	Sometimes for invalid receiver configuration combinations, Ultra Messaging repeats warning log messages for every transport joined. As a solution, invalid receiver configuration combinations now issue this warning once at receiver creation time.
8608	A Java or .NET hot failover receiver that experiences loss might assert with fatal assertion↵ :[d_entry->topic_entry != NULL].
8610	Message properties do not clear in Java and C# before loading the properties of the next message to be delivered.
8633	The function lbm_context_process_events() returns -1 for non-critical errors without further description. As a solution, Ultra Messaging no longer returns -1 from lbm_context ↵ process_events() for non-critical errors.
8662	In the Java API, the array of stores returned by com::latencybusters::lbm::LBMSource ↵ Attributes::getStores() contains an incorrect number of entries when using named stores. In addition to the solution for this, the com::latencybusters::lbm::UMESourceEntry class now contains a new field to provide store name information.

Change Request	Description
8682	<p>The lbm_serialize_response() function does not include the Domain ID in the serialized message. When a serialized response message traverses a UM Router (DRO) to a different topic resolution domain, the message cannot reach the requester, and the lbm_deserialize_response() function logs the following warning:</p> <pre>Core-6259-25: Deserialize Response: Context in domain 1 received response with no domain...</pre> <p>As a solution, serialized response messages now include the topic resolution domain ID.</p>
8726	<p>If you set ume_retention_unique_confirmations (source) to 0 and ume_confirmed_delivery_notification (source) to 3, the first delivery confirmation sets the UNIQUE_ACKS flag. As a solution, this flag now sets only if you set option ume_retention_unique_confirmations (source) to a value greater than 0 and the number of confirmations received are enough to exceed the threshold.</p>
8732	<p>With option hf_duplicate_delivery (receiver) and ordered delivery enabled, hot failover receivers might deliver duplicate messages before the original messages. With this solution, messages flagged as duplicate are always be delivered after messages not flagged as duplicate.</p>
8733	<p>Hot failover receivers might leak memory when processing HF reset commands or when deleted.</p>
8743	<p>An Ultra Messaging application or daemon fatally asserts with message:</p> <pre>[dest] at line 481 in ../../../../src/lib/lbm/lbmrxrctlr.c</pre> <p>if the application receives malformed data on an Ultra Messaging tcp socket or under other rare conditions.</p>
8797	<p>Java method com::latencybusters::lbm::LBM::setDebugMask() sets incorrect mask values on 32-bit platforms.</p>
8868	<p>In the Java and .NET APIs, some objects accessible from an LBMMMessage object are sometimes newly created for each LBMMMessage delivered, in violation of the Zero Object Delivery (ZOD) functionality. The affected objects include com::latencybusters::lbm::LBMMMessageChannelInfo for messages received on a spectrum channel, com::latencybusters::lbm::UMQIndexInfo for queueing messages with an index, and com::latencybusters::lbm::UMQMessageId for all queueing or ULB messages received. As a solution, com::latencybusters::lbm::LBMMMessage objects now re-use these object types for each new message delivery, rather than creating new objects.</p> <p>Note</p> <p>You might have applications that no longer work correctly because of this change. To use the 'LBMMMessageChannelInfo', 'UMQIndexInfo', or 'UMQMessageId' objects outside of the onReceive callback, you must first make a copy of the object using its constructor, or call promote() on the LBMMMessage object.</p>

16.2.2 Persistence Fixed Limitations for 6.7

The following bug fixes apply to UMP and UMQ products.

Change Request	Description
1013	When a umestored configuration file sets the '<log>' type attribute to the unsupported value of syslog, the umestored daemon reports the following error message: daemon section log entity must have a value As a solution, the umestored daemon now reports the following error message: daemon log does not currently support syslog type
3894	When both registration IDs and session IDs are present, Ultra Messaging issues a log message stating that registration IDs are being ignored, but then fails to ignore them.
5273	Upon startup, a store could delay persistence of messages for up to the configured retransmit_request_generation_interval value, while attempting to recover messages from the source that the source has already reclaimed. As a solution, the source reports the non-availability of the requested messages back to the store, and the store can begin persistence without extra delay.
5498	The umestored daemon warns that option resolver_active_threshold (context) is deprecated. This warning is now removed. Setting this option continues to have no effect.
6112	The umercv sample application prints session IDs in decimal format. As a resolution, all sample applications now print Session IDs in hexadecimal, and continue to print registration IDs in decimal.
7282, 8223	A store might fail in certain incorrect registration scenarios. As a solution, if applications present invalid session ID and registration ID combinations, they now receive a registration error.
8171	An RPP-configured source can get an assert with the following log message: LOG Level 1: FATAL: failed assertion [info->sqn==ctrl->trail← _sqn] at line 305 in ../../../../src/lib/lbm/lbmrxsctrl.c This can happen when the source re-registers with a reduced-fd repository.
8185	RPP stores can now send a stability ack for a message that was lost but later recovered via retransmission.
8245	Configuring UMP stores using non-increasing group index order, such as configuring storeA in group 2 before storeB in group 1, causes erroneous registration changes and indeterminate receiver behavior.
8677	With the Java API, on a Persistence deregistration event, the Ultra Messaging library delivers a com::latencybusters::lbm::UMEDeregistrationCompleteInfo with invalid contents. As a solution, since there is no information associated with this event, the com::latencybusters::lbm::LBMessage::deregistrationCompleteInfo() method now always returns null.
8704	Ultra Messaging handles stability acknowledgements on the same thread as retransmissions, affecting the consistency of latency. As a solution, stability acknowledgements now use a different thread.
8710	With option ume_confirmed_delivery_notification (source) set to 3, after ume_← retention_unique_confirmations (source) is reached, the whole message confirmed flag is not set for delivery confirmations for all receivers.
8711	In the .NET (C#) API, there are no methods to access objects com::latencybusters::lbm← ::UMESourceEventDeregistrationSuccessInfo and com::latencybusters::lbm::UME← SourceEventDeregistrationCompleteInfo . As a solution, the C# library now includes accessor methods.

Change Request	Description
8747	Sometimes an application receives and acts upon retransmit response information intended for a different application. This solution makes retransmit requests more unique to the application issuing the request to prevent potential cross-contamination of retransmit response information during application restarts.
8766	UMP stores might accept messages from a UMP source that has not completed the registration process, resulting in the following fatal assertion: [EMERGENCY]: FATAL: failed assertion [MUL_SQN_↵ GTE(info->sqn, next_avail_sqn)] at line 1221 in ../../../../src/stored/umerepo.c
8804	If there is a disk error modifying the state file on receiver registration, the umestored daemon writes to an invalid location, possibly causing a corrupted state file or umestored segmentation fault. As a solution, umestored no longer writes to an invalid location but instead issues a log-message warning.
8808	Store daemons sometimes issue a Store-5688-5258 message. As a solution, the umestored configuration code now disallows using multiple async write callbacks. This prevents situations where multiple writes could be executed out of order, causing data corruption.

16.2.3 Queuing Fixed Limitations for 6.7

Note

The brokered queuing and UM JMS functionalities were replaced in version 6.8, so most of these notes do not apply to more-recent versions of UM. However, ULB functionality is retained.

The following bug fixes apply to the UMQ product.

Change Request	Description
8315	In the .NET API, Ultra Messaging reuses com::latencybusters::lbm::LBMessage↵ ChannelInfo , com::latencybusters::lbm::UMQMessageId , and com::latencybusters↵ ::lbm::UMQIndexInfo objects between different com::latencybusters::lbm::LBM↵ Message objects for delivery, reducing garbage collection overhead.
8325	ULB reassigns messages even if option umq_ulb_application_set_message_max↵ reassignments (source) is set to 1.
8536	When restarting the UMQ daemon while handling zero-length-messages, the daemon sometimes fails with the following fatal assertion: [EMERGENCY]: FATAL: failed assertion [msg->lbm_msg!=NULL] at line 3896 in ../../../../src/stored/umqsinc.c
8601	An Ultra Messaging application crashes when ULB or UMQ receivers send a Consumption Report for a message after their source transport had been torn down.

16.2.4 Dynamic Router Fixed Limitations for 6.7

The following bug fixes apply to the Dynamic Routing Option (DRO).

Change Request	Description
7276	When using an aggressive keepalive configuration on a UM Router (DRO) peer link, the UM Router might fatally terminate.
8256	When configuring a UM Router (DRO) Peer Connection on a Microsoft Windows machine, the single-tcp configuration option can result in no message traffic across the link. As a solution, Ultra Messaging now handles all documented error conditions, instead of only 'WSAECONNREFUSED'. This solution addresses issues where a UM Router would not re-connect with a peer under timeout conditions.
8303	A UM Router (DRO) might crash when peer connections between two UM Routers quickly disconnect and reconnect.
8446	For Windows and 32-bit UNIX platforms, the web monitor displays non-zero values for the statistic "<i>Transport topic fragments/bytes dropped due to blocking</i>", but Ultra Messaging does not log the message [information] Gwd-6945-2: Portal [xxx] dropping data due to high volume.
8459	Single TCP connections repeatedly disconnect and reconnect under certain conditions.
8461	UM Router (DRO) peer connections, both dual and single TCP, might encounter an error and not be restarted. One side effect for the solution to this is that the UM Router now logs every connection attempt failure, resulting in more log messages if the other peer is not up.
8752	Error message: Core-5688-1890: handle events returned error 4 [Core← Api-5688-3837: WFMO: (87) The parameter is incorrect] sometimes occurs, primarily in a peer portal connection.
8763	The UM Router (DRO) sometimes asserts with the messages: WARNING: failed assertion [Portal->peer.adjacent_node_← id==0] at line 2679 in ../../../../src/gateway/tnwgpeer.c and: FATAL: failed assertion [void_o_entry!=NULL] at line 4205 in ../../../../src/gateway/tnwgpeer.c

16.3 Deprecations for 6.7

16.3.1 Streaming Deprecations for 6.7

- The `lbmmon_attr_get_contextid()` function is deprecated. Instead, use `lbmmon_attr_get_objectid()`.

16.3.2 Persistence Deprecations for 6.7

- The Round-Robin Store failover configuration is deprecated.

- The 'repository-disk-max-write-async-cbs' option is deprecated.

Chapter 17

UM Version 6.5

The most-significant update to UM version 6.5 is the introduction of the Ultra Messaging System Monitoring option.

Warning

UM version 6.5 does not interoperate reliably with versions of UM prior to 6.0. It was not until UM version 6.7 and beyond that interoperability was introduced between the 6.x version line and the 5.x version line.

17.1 Enhancements for 6.5

17.1.1 Streaming Enhancements for 6.5

The following new features and enhancements apply to UMS, UMP, and UMQ products.

Change Request	Description
7911	You can now configure Ultra Messaging to use Solarflare Onload to select transport sockets for acceleration. Onload is a kernel-bypass technology that accelerates message traffic and operates with Solarflare 10GbE Ethernet NICs. See Solarflare® Onload for more information.
7974	The lbmrd daemon now includes command line option '-f' or '--lic-file' to let you specify the Ultra Messaging license file.

17.1.2 Persistence Enhancements for 6.5

The following new features and enhancements apply to UMP and UMQ products.

Change Request	Description
7478	The Persistent Store Web Monitor now displays the Session ID on the Store page, Source page, and Receiver page.

17.1.3 Queuing Enhancements for 6.5

Note

The brokered queuing and UM JMS functionalities were replaced in version 6.8, so most of these notes do not apply to more-recent versions of UM. However, ULB functionality is retained.

The following new features and enhancements apply to the UMQ product.

- None.

17.1.4 Dynamic Router Enhancements for 6.5

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

Change Request	Description
7694	You can now use the LBT-SMX transport to send ingress message traffic to a UM Router (DRO). The UM Router no longer ignores topic resolution traffic from a LBT-SMX source. You still cannot, however, use LBT-SMX to send egress traffic from a UM Router.

17.2 Ultra Messaging System Monitoring Option

In conjunction with the Ultra Messaging 6.5 release, Informatica has introduced the Ultra Messaging System Monitor Option (UMSM). Administrators can use UMSM to monitor components of an Ultra Messaging deployment. The UMSM dashboard displays the status of components, such as UM hosts, transports, topic resolution domains, and application instances, as well as UM statistics.

UMSM monitors the messaging traffic of Ultra Messaging versions 5.0 and later.

With the Ultra Messaging 6.5 release, Ultra Messaging Automatic Monitoring sends the following aggregated receiver statistics and subscription information to UMSM:

- Aggregate number of topic-level sequence numbers received out of order.
- Aggregate number of topic-level sequence numbers declared unrecoverably lost by the delivery of either an **LBM_MSG_UNRECOVERABLE_LOSS** or **LBM_MSG_UNRECOVERABLE_LOSS_BURST** event to the receiving application.
- Receiver callback service time statistics. Ultra Messaging tracks the mean time, maximum time, and minimum time to service receiver callbacks.

As the collection of aggregated statistics can increase the latency of message delivery, Ultra Messaging does not collect them by default. You can enable UMSM to collect the aggregated message loss statistics with the UM configuration options, **monitor_interval (receiver)** and **monitor_interval (wildcard_receiver)**. You can access the aggregated receiver statistics with UMSM. The lbmmon API does not provide access to the aggregated receiver statistics.

The Ultra Messaging 6.5 release contains a third new UM configuration option, **receiver_callback_service_time**_↔**_enabled (context)**, which enables UMSM to maintain receiver callback service times. The lbmmon API does not provide access to the receiver callback service times.

17.3 Fixed Limitations for 6.5

17.3.1 Streaming Fixed Limitations for 6.5

The following bug fixes apply to UMS, UMP, and UMQ products.

17.3.2 Persistence Fixed Limitations for 6.5

The following bug fixes apply to UMP and UMQ products.

17.3.3 Queuing Fixed Limitations for 6.5

Note

The brokered queuing and UM JMS functionalities were replaced in version 6.8, so most of these notes do not apply to more-recent versions of UM. However, ULB functionality is retained.

The following bug fixes apply to the UMQ product.

17.3.4 Dynamic Router Fixed Limitations for 6.5

The following bug fixes apply to the Dynamic Routing Option (DRO).

Chapter 18

UM Version 6.1

The most-significant update to UM version 6.1 is the introduction of the LBT-SMX transport type. See [New Transport LBT-SMX](#).

Warning

UM version 6.1 does not interoperate reliably with versions of UM prior to 6.0. It was not until UM version 6.7 and beyond that interoperability was introduced between the 6.x version line and the 5.x version line.

18.1 Enhancements for 6.1

18.1.1 Streaming Enhancements for 6.1

The following new features and enhancements apply to UMS, UMP, and UMQ products.

Change Request	Description
	Added LBT-SMX
7529	Added the polling of context events when using ibm_context_process_events() with operational_mode (context) set to <i>sequential</i> . You can now set the ibm_context_↵_process_events() parameter 'msec' to 0 (zero), which results in the UM context making one pass to process timers, sockets and then timers again before returning the thread to the application without any delay. (Previously, if you set the msec parameter to 10 and the UM context took 5 msec to process all events, it would wait another 5 msec before returning the thread to the application.)
7664	Ultra Messaging no longer sends a Topic Management Record (TMR) when an application deletes a receiver in a standard way with, for example ibm_rcv_delete() , when Dynamic Routers do not exist in the Topic Resolution domain. TMRs specifically inform a Dynamic Router that a receiver for a particular topic has been shutdown. (Without the presence of Dynamic Routers, TMRs become unnecessary network traffic.)
7551	Added SIGTERM to the list of signals that initiate clean shutdown of all Ultra Messaging daemons.

Change Request	Description
4785	Added the ability in the Java and .NET APIs to restrict receiver objects to a single LBM↔ ReceiverCallback with a new API: com::latencybusters::lbm::LBMReceiverAttributes↔ ::enableSingleReceiverCallback() . You can reduce latencies and increase throughput by using a single callback which removes the need for any synchronization of multiple call-backs within UM during message delivery.
7752	The Java API now supports Java version 1.7.

18.1.2 New Transport LBT-SMX

The LBT-SMX (shared memory acceleration) transport is an Interprocess Communication (IPC) transport you can use for the lowest latency message streaming. LBT-SMX is faster, though more-restrictive than the LBT-IPC transport. Like LBT-IPC, sources can publish topic messages to a shared memory area from which receivers can read topic messages. Unlike LBT-IPC, the native APIs for the LBT-SMX transport are not thread safe and do not support all UM features such as message batching or fragmentation.

You can use either the native LBT-SMX API calls, **lbm_src_buff_acquire()** and **lbm_src_buffs_complete()** to send over LBT-SMX or you can use **lbm_src_send_***() API calls. The existing send APIs are thread safe with SMX, but they incur a synchronization overhead and thus are slower than the native LBT-SMX API calls.

LBT-SMX operates on the following Ultra Messaging 64-bit packages:

- SunOS-5.10-amd64
- Linux-glibc-2.5-x86_64
- Win2k-x86_64

The example applications **lbmlatping.*** and **lbmlatpong.*** show how to use the native LBT-SMX API calls. All three APIs (C API, Java API, and .NET API) have identically named example applications. Other example applications can use the LBT-SMX transport with the use of a UM configuration file containing source transport **lbtsmx**. You cannot use LBT-SMX with example applications for features not supported by LBT-SMX, such as **lbmreq.***, **lbmresp.***, **lbmrcvq.*** or **lbmwrcvq.***.

The LBT-SMX configuration options are similar to the LBT-IPC transport options. See Transport **Transport LBT-S↔ MX Operation Options**.

You can use Automatic Monitoring, UM API retrieve/reset calls, and LBMMON APIs to access LBT-SMX source and receiver transport statistics. To increase performance, the LBT-SMX transport does not collect statistics by default. Set the UM configuration option **transportlbtsmxmessagestatisticsenabledcontext** to 1 to enable the collection of transport statistics.

18.1.3 Persistence Enhancements for 6.1

The following new features and enhancements apply to UMP and UMQ products.

- None.

18.1.4 Queuing Enhancements for 6.1

Note

The brokered queuing and UM JMS functionalities were replaced in version 6.8, so most of these notes do not apply to more-recent versions of UM. However, ULB functionality is retained.

The following new features and enhancements apply to the UMQ product.

- None.

18.1.5 Dynamic Router Enhancements for 6.1

The following new features and enhancements apply to the Dynamic Routing Option (DRO).

- None.

18.2 Fixed Limitations for 6.1

18.2.1 Streaming Fixed Limitations for 6.1

The following bug fixes apply to UMS, UMP, and UMQ products.

Change Request	Description
7656	Fixed a problem with <code>lbm_set_umm_info()</code> that caused it to always return 0 (success). The function now returns -1 on a parsing error of a returned UMM configuration file, UMM Daemon login error or a communications error with the UMM Daemon.

18.2.2 Persistence Fixed Limitations for 6.1

The following bug fixes apply to UMP and UMQ products.

Change Request	Description
7606	When a receiver tries to register with a store using a RegID in use by another receiver, the application error log message now includes the RegID: <code>UME registration error: error in UME registration, RegID 1508542809 is in use by a different receiver at store 0</code> Previously the message always displayed RegID 0 and no store number.

Change Request	Description
7604 7666	When a source tries to register with a store using a RegID in use by another source, the application error log message now includes the RegID, store number and the store's IP address and port: 'Error registering source with UME store: error in UME registration, RegID 2323214065 is in use by a different source at store 1 192.168.101.130.21112' Previously the message always displayed RegID 0 and no store information.
7543	Corrected a problem with UMP stores that resulted in stability acknowledgements sent from a persistent store to the source during a store proxy source election contained invalid topic and transport indexes. As a result, the source discarded the stability acknowledgements.
7658	The umestored Web Monitor has been changed to correctly display the Request Receive Rate, Service Rate and Drop Rate. Previously the rates only showed activity for the last second instead of a true activity per second rate. As a result the rates often displayed 0 (zero). A link has been added to the Persistent Store Page to reset the rate calculation. The rates can be reset on a per store basis.
7262	When a store receives a retransmission request from an unknown source, the error log message now includes the IP address and port number of the unknown source: Core-5688-696: received retransmit request for unknown source 192.168. 101.112:45512

18.2.3 Queuing Fixed Limitations for 6.1

Note

The brokered queuing and UM JMS functionalities were replaced in version 6.8, so most of these notes do not apply to more-recent versions of UM. However, ULB functionality is retained.

The following bug fixes apply to the UMQ product.

Change Request	Description
7545	Corrected an issue that prevented a UMQ receiver for a dead letter topic from registering with a queue if any queue instance did not have a record of the creation of the dead letter topic.

18.2.4 Dynamic Router Fixed Limitations for 6.1

The following bug fixes apply to the Dynamic Routing Option (DRO).

- None.

Chapter 19

UM Version 6.0

The most-significant update to UM version 6.0 is the introduction of the Dynamic Routing Option (DRO). The UM Router component fully replaces the "Gateway" component, which is discontinued as of UM 6.0. See the [Dynamic Routing Guide](#) for full information.

Warning

UM version 6.0 does not interoperate reliably with any previous version of UM. It was not until UM version 6.7 and beyond that interoperability was introduced between the 6.x version line and the 5.x version line.

19.1 Enhancements for 6.0

19.1.1 Streaming Enhancements for 6.0

The following new features and enhancements apply to UMS, UMP, and UMQ products.

- The receiver configuration options **retransmit_request_generation_interval (receiver)** and **otr_request_duration (receiver)** have been deprecated. Late join and OTR now track a total duration for each message that begins when a receiver sends the first retransmission request for the message. See **retransmit_request_message_timeout (receiver)**.
- Linux platforms now automatically produce a back trace listing to the log file upon encountering a fatal assertion.
- Added the following configuration options to help control receiver start-up traffic:
 - **late_join_info_request_interval (receiver)**
 - **late_join_info_request_maximum (receiver)**
 - **transport_topic_sequence_number_info_request_interval (receiver)**
 - **transport_topic_sequence_number_info_request_maximum (receiver)**
- Network Address Translation with the Unicast Resolver Daemon (lbmrdr) using LBT-RU with Session IDs is now enabled. Previously this combination only worked with Session IDs disabled.
- The context setting **network_compatibility_mode (context)** has been disabled. Specifying it does not cause an error, but it will be ignored.

- Added TCP Session IDs to prevent the use of outdated data in the Topic Resolution Cache. You enable session ids for TCP with the configuration option **transport_tcp_use_session_id (source)**.
- Removed a fatal assert that was produced if a message was delivered from a TSNI request.
- The Off-Transport Recovery (OTR) feature has been updated to include a new option **otr_message_caching_threshold (receiver)** and to use option **retransmit_message_caching_proximity (receiver)**, both of which provide ways to control receiver buffer growth and otherwise premature unrecoverable loss declarations. Also, option **otr_request_duration (receiver)** is deprecated and replaced with the new option **otr_request_message_timeout (receiver)**.
- The Late Join option **retransmit_request_generation_interval** is deprecated and replaced with new option **retransmit_request_message_timeout (receiver)**. The new option defines an RX give-up interval on a per-message basis.

19.1.2 Persistence Enhancements for 6.0

The following new features and enhancements apply to UMP and UMQ products.

- Proactive Retransmissions - This feature address two types of loss:
 - Loss of message data between the source and a store.
 - Loss of stability acknowledgments (ACK) between the store and the source.

See Proactive Retransmissions in Chapter 8 of the UM Guide for Persistence. This feature also resolves the Known Issue: *"UMP stores sometimes fail to send message stability acknowledgements to sources for messages sent while the store was being reported unresponsive by the source."*
- The persistence registration process has been improved for greater reliability, efficiency and to better inform receivers of store registration changes. Persistence now sends store registration information over the transport and not via Topic Resolution advertisements. Several new configuration options have been added to tune registration latency/bandwidth usage.
- Added the following configuration options to help control receiver start-up traffic.
 - **ume_sri_request_interval (receiver)**
 - **ume_sri_request_maximum (receiver)**
- The **ume_store (source)** configuration option now accepts an optional DomainID parameter.
- Made context name resolution more efficient. Several new UM configuration options have been added to tune context name resolution. Context name resolution now has the potential to quiesce over time. New log messages have been added to facilitate duplicate context name detection. For more information, see:
 - **Store (context) Name Resolution**
 - **Identifying Persistent Stores**
- The store option **retransmission-request-forwarding** is now interpreted differently. If enabled (value = 1), the store forwards retransmission requests to sources (if and only if it does not have the data). If disabled (value = 0), the store never forwards requests to sources.
- Enhanced the proxy source election process in the following ways:
 - Proxy source elections can now occur among stores located in different topic resolution domains when using the UM Router (DRO).
 - The election now occurs as part of the registration process instead of over the topic resolution channel, so it is more bandwidth efficient.

- The election process no longer restarts in the event of a tie, so it will always complete in a deterministic amount of time.

This update also resolves the Known Issue: "*Persistence proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links.*"

- The persistent store no longer logs Beginning of Session (BOS) and End of Session (EOS) events.
- When a store restarts, the store now renames and stores state and cache files that are thought to be corrupted instead of deleting them. These files will be renamed from '`<regid>-<state/cache>`' to '`<regid>-<state/cache>.<pid>.<month_day_year>`'. Persistent stores the files in the same cache/state directories as specified in the store's configuration file.
- A rare error case that occurred when handling a keepalive at the persistent store now logs a warning instead of fatal assert.
- Changed the Persistent Store (umestored) to ignore UM Spectrum configuration options which results in the delivery of all messages to the store without any filtering. See **Spectrum**.

19.1.3 Queuing Enhancements for 6.0

Note

The brokered queuing and UM JMS functionalities were replaced in version 6.8, so most of these notes do not apply to more-recent versions of UM. However, ULB functionality is retained.

The following new features and enhancements apply to the UMQ product.

- Added support in Ultra Messaging JMS for store names in the JMSConfig.xml so that UM JMS can be configured to work with the UM Router (DRO) for UMS/UMP functionality.
- Wildcard topics can now be created programmatically in Ultra Messaging JMS. Use '`<FactoryAttributes>`' in either a UM Configuration File or the JMSConfig.xml file support this new feature.
 - `wildcards_enabled`
 - `wildcard_type`
 - `wildcard_enterprise_mode` - Provides compatibility with other JMS vendors. If the topic name contains "*" or ">", UM JMS creates a PCRE wildcard topic pattern. For example, the topic 'PRICE.STOCK.*.IBM' creates a wildcard receiver with a topic pattern of '`^PRICE\ .STOCK\ . [^/]+\ .IBM$`'.
 - `wildcard_um_values` - If the topic name contains any of the characters:
`[\ ^ $. | ? + () { } *]`
 UM JMS creates a PCRE wildcard topic pattern.
 - The option '`wildcard_enterprise_wildcard_delimiter`' supports any character as a delimiter except:
`[\ ^ $. | ? + () { } *]`
- Changed the "wait for source registration" Ultra Messaging JMS functionality to apply to send calls rather than source creation. The two relevant config options (`wait_for_source_registration` and `SOURCE_REGISTRATION_TIMEOUT`) are no longer considered during source creation but instead cause a producer to (optionally) have a send call block until source registration occurs. This change removes the need for the producer to explicitly handle the "not registered" exception that would otherwise be thrown and need to be handled by the application. The existing Ultra Messaging JMS configuration option '`SOURCE_CREATION_DELAY`' has not been changed.
- The UM JMS source code examples have been moved out of the '`<platform>/jmsclient/bin`' directory and up to the '`<platform>/example`' directory.

- Changed the store names in the JMS config.xml file from uJMSSStore1, uJMSSStore2 and uJMSSStore3 to JMSSStore1, JMSSStore2 and JMSSStore3. Also renamed the Queue from uJMSQueue to JMSQueue.

19.1.4 Dynamic Router Enhancements for 6.0

UM 6.0 is the first version which offers the Dynamic Routing Option (DRO). See the [Dynamic Routing Guide](#) for full information.

19.2 Fixed Limitations for 6.0

19.2.1 Streaming Fixed Limitations for 6.0

The following bug fixes apply to UMS, UMP, and UMQ products.

- Corrected a problem with UM rate controls that resulted in rate limits being applied unequally among more than one transport session. Limits are now applied equally to all active transport sessions.
 - Changed the implementation of the default value for **mim_sqn_window_increment (context)** to match the documented value of 8192, which is the recommended default.
 - Corrected a problem with plain text (non-XML) UM configuration files greater than 1024 bytes that when loaded via a URL (http or FTP) would result in line malformed errors. This fix resolves the Known Issue: *"Retrieving plain text (not XML) configurations files via an http or FTP request is limited to configuration files of 1K in size..."*
 - Fixed a problem on Microsoft Windows where an interface could be incorrectly chosen internally as the default interface even though its operational status was not "up".
 - Corrected an invalid single-byte memory write that could occur if passing in a named interface setting into any of the *_interface configuration options.
 - Corrected calls to socket 'recv/recvfrom' that were not correctly handling EWOULDBLOCK/EAGAIN errors. This has been resolved.
 - The following Known Issue was resolved in UM 5.3.1 but mistakenly omitted from the list of Bug Fixes: "<i>↵ The .NET API does not support message selectors with Unicode symbols and causes receiver creation to fail.</i>"
 - Fixed a problem that could cause a crash while binding TCP ports.
 - Fixed a '[msg==NULL]' fatal assert that could occur if an application received a response to a Late Join Information request that contained a message, and not Late Join Information.
 - Corrected an issue with an event queue that could contain events, but never be dispatched. This issue could occur only on the AIX and NonStop platforms.
 - Corrected a problem with receivers that produced Unrecoverable Loss if UM delivered a TSNI to the receiver during Late Join recovery.
-

19.2.2 Persistence Fixed Limitations for 6.0

The following bug fixes apply to UMP and UMQ products.

- Fixed a problem that logs a keep alive sent from a store during UMP source de-registration as an error. This event no longer logs an error in the store log file.
- Fixed a condition that could cause sources with high `retransmit_request_size_threshold/limit` settings to experience high CPU usage when a store daemon (umestored) was restarted.
- Fixed a problem that could cause a crash if a source tried to de-register from a store with which it had never registered or whose context name had not been resolved.
- Fixed the incorrect retention of the receiver-specified **ume_activity_timeout (source)** and **ume_state_lifecycle (receiver)** values across store restarts. The receiver-activity-timeout configured in the stores' XML configuration file was used instead and the receiver-state-lifetime was ignored. Stores now correctly save receivers' configured activity timeout and state lifetime and use the saved values upon store restart.
- Resolved the Known Issue: "*The UMP store daemon, umestored, stops persisting messages to disk if the store has a loss condition which is not resolved prior to an EOS event...*"
- Corrected a problem that caused segfaults during long running store recoveries by delaying the store's check of sources if the store has not yet finished initializing.
- Fixed an issue that occurred if the store index was uninitialized when a de-registration success event was delivered.
- Fixed a bug that could prevent a receiver from successfully deregistering from a store.
- Corrected a problem that prevented receivers from declaring completed deregistration although the receiver successfully deregistered from all the stores with which it was registered. Now a receiver declares completed deregistration as soon as it receives deregistration success from all the stores with which it was registered.
- Corrected a number of problems with UMP Proxy Source Elections that resulted in numerous failed assertion warnings and the continuous creation and duplication of proxy sources.
- Corrected a problem with source re-registration to a store that caused the store to incorrectly log warning messages such as, "*<i>Core-5688-1881: handle events returned error 4</i>*" and "*<i>CoreApi-5480-28: could not insert rcvdc_loss_rec ASL map.</i>*"
- Corrected a problem with the persistent store operating in RPP mode that caused a fatal assert if the store restarted with a prematurely ending cache file.
- Corrected a problem that caused a store to perpetually send TSNI requests to a source at the configured **retransmit_request_interval (receiver)**, even after live data was seen on the transport.
- Fixed a problem with sending large messages using message properties that could cause a crash due to message reclaim source events.
- Corrected a problem with the umestored daemon that resulted in the premature deletion of source state information if umestored was restarted. If the source had an active receiver before umestored was shut down, the configured **source-state-lifetime** was not restored, so the default lifetime of 0 seconds and default **source-activity-timeout** of 30 seconds was enforced.

19.2.3 Queuing Fixed Limitations for 6.0

Note

The brokered queuing and UM JMS functionalities were replaced in version 6.8, so most of these notes do not apply to more-recent versions of UM. However, ULB functionality is retained.

The following bug fixes apply to the UMQ product.

- Corrected a problem with UM JMS that caused a `JMSException` when using lower case variants of the 'ConnectionFactory' attributes `receiver_creation_delay`, `source_creation_delay`, and `source_registration_timeout` in a `config.xml` file. You can now enter these ConnectionFactory attributes in lower case.
- Resolved a problem in UM JMS that caused an exception and a loss of messages from a queue when restarting a receiver. Messages unacknowledged by a receiver that shuts down are now put back onto the queue. These messages are available for resending to the receiver when it restarts.
- Fixed a problem with the UMQ daemon (`umestored`) that could lead to a crash if you kept queue browsing support enabled (default) and multiple queue instances are rapidly killed and restarted in sequence. The crash would only potentially occur if log messages appeared in the `umestored` log file indicating that a message "was already present in SINC msg list".
- Fixed a rare bug that could lead to a crash in the UMQ daemon (`umestored`) when configured to use a disk and many sending threads, which is not the default.

19.2.4 Dynamic Router Fixed Limitations for 6.0

The following bug fixes apply to the Dynamic Routing Option (DRO).

- None.

19.3 Deprecations for 6.0

19.3.1 Streaming Deprecations for 6.0

- The receiver configuration options **`retransmit_request_generation_interval (receiver)`** and **`otr_request_duration (receiver)`** have been deprecated. Late join and OTR now track a total duration for each message that begins when a receiver sends the first retransmission request for the message. See **`retransmit_request_message_timeout (receiver)`**.
 - Option **`otr_request_duration (receiver)`** is deprecated and replaced with the new option `otrrequestmessage-timeoutreceiver`.
 - The Late Join option `retransmit_request_generation_interval` is deprecated and replaced with new option **`retransmit_request_message_timeout (receiver)`**. The new option defines an RX give-up interval on a per-message basis.
-