

# UMS / UMP / UMQ Change Log

## Informatica® Corporation

Copyright © 2004 - 2014 Informatica Corporation. All rights reserved.  
March 2014

Informatica Ultra Messaging  
Version 5.3  
March 2014

Copyright (c) 1998-2014 Informatica Corporation. All rights reserved.

This software and documentation contain proprietary information of Informatica Corporation and are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright law. Reverse engineering of the software is prohibited. No part of this document may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording or otherwise) without prior consent of Informatica Corporation. This Software may be protected by U.S. and/or international Patents and other Patents Pending.

Use, duplication, or disclosure of the Software by the U.S. Government is subject to the restrictions set forth in the applicable software license agreement and as provided in DFARS 227.7202-1(a) and 227.7702-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (OCT 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14 (ALT III), as applicable.

The information in this product or documentation is subject to change without notice. If you find any problems in this product or documentation, please report them to us in writing.

Informatica, Informatica Platform, Informatica Data Services, PowerCenter, PowerCenterRT, PowerCenter Connect, PowerCenter Data Analyzer, PowerExchange, PowerMart, Metadata Manager, Informatica Data Quality, Informatica Data Explorer, Informatica B2B Data Transformation, Informatica B2B Data Exchange Informatica On Demand, Informatica Identity Resolution, Informatica Application Information Lifecycle Management, Informatica Complex Event Processing, Ultra Messaging and Informatica Master Data Management are trademarks or registered trademarks of Informatica Corporation in the United States and in jurisdictions throughout the world. All other company and product names may be trade names or trademarks of their respective owners.

Portions of this software and/or documentation are subject to copyright held by third parties, including without limitation: Copyright DataDirect Technologies. All rights reserved. Copyright (c) Sun Microsystems. All rights reserved. Copyright (c) RSA Security Inc. All Rights Reserved. Copyright (c) Ordinal Technology Corp. All rights reserved. Copyright (c) Aandacht c.v. All rights reserved. Copyright (c) Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright (c) Meta Integration Technology, Inc. All rights reserved. Copyright (c) Intalio. All rights reserved. Copyright (c) Oracle. All rights reserved. Copyright (c) Adobe Systems Incorporated. All rights reserved. Copyright (c) DataArt, Inc. All rights reserved. Copyright (c) ComponentSource. All rights reserved. Copyright (c) Microsoft Corporation. All rights reserved. Copyright (c) Rogue Wave Software, Inc. All rights reserved. Copyright (c) Teradata Corporation. All rights reserved. Copyright (c) Yahoo! Inc. All rights reserved. Copyright (c) Glyph & Cog, LLC. All rights reserved. Copyright (c) Thinkmap, Inc. All rights reserved. Copyright (c) Clearpace Software Limited. All rights reserved. Copyright (c) Information Builders, Inc. All rights reserved. Copyright (c) OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved. Copyright Cleo Communications, Inc. All rights reserved. Copyright (c) International Organization for Standardization 1986. All rights reserved. Copyright (c) ej-technologies GmbH. All rights reserved. Copyright (c) Jaspersoft Corporation. All rights reserved. Copyright (c) is International Business Machines Corporation. All rights reserved. Copyright (c) yWorks GmbH. All rights reserved. Copyright (c) Lucent Technologies. All rights reserved. Copyright (c) University of Toronto. All rights reserved. Copyright (c) Daniel Veillard. All rights reserved. Copyright (c) Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright (c) MicroQuill Software Publishing, Inc. All rights reserved. Copyright (c) PassMark Software Pty Ltd. All rights reserved. Copyright (c) LogiXML, Inc. All rights reserved. Copyright (c) 2003-2010 Lorenzi Davide, All rights reserved. Copyright (c) Red Hat, Inc. All rights reserved. Copyright (c) The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright (c) EMC Corporation. All rights

reserved. Copyright (c) Flexera Software. All rights reserved. Copyright (c) Jinfonet Software. All rights reserved. Copyright (c) Apple Inc. All rights reserved. Copyright (c) Telerik Inc. All rights reserved. This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>), and/or other software which is licensed under various versions of the Apache License (the "License"). You may obtain a copy of these Licenses at <http://www.apache.org/licenses/>. Unless required by applicable law or agreed to in writing, software distributed under these Licenses is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the Licenses for the specific language governing permissions and limitations under the Licenses.

This product includes software which was developed by Mozilla (<http://www.mozilla.org/>), software copyright The JBoss Group, LLC, all rights reserved; software copyright (c) 1999-2006 by Bruno Lowagie and Paulo Soares and other software which is licensed under various versions of the GNU Lesser General Public License Agreement, which may be found at <http://www.gnu.org/licenses/lgpl.html>. The materials are provided free of charge by Informatica, "as-is", without warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability and fitness for a particular purpose.

The product includes ACE(TM) and TAO(TM) software copyrighted by Douglas C. Schmidt and his research group at Washington University, University of California, Irvine, and Vanderbilt University, Copyright (c) 1993-2006, all rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (copyright The OpenSSL Project. All Rights Reserved) and redistribution of this software is subject to terms available at <http://www.openssl.org> and <http://www.openssl.org/source/license.html>.

This product includes Curl software which is Copyright 1996-2007, Daniel Stenberg, <[daniel@haxx.se](mailto:daniel@haxx.se)>. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://curl.haxx.se/docs/copyright.html>. Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

The product includes software copyright 2001-2005 (c) MetaStuff, Ltd. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://www.dom4j.org/license.html>.

The product includes software copyright (c) 2004-2007, The Dojo Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://dojotoolkit.org/license>.

This product includes ICU software which is copyright International Business Machines Corporation and others. All rights reserved. Permissions and limitations regarding this software are subject to terms available at <http://source.icu-project.org/repos/icu/icu/trunk/license.html>.

This product includes software copyright (c) 1996-2006 Per Bothner. All rights reserved. Your right to use such materials is set forth in the license which may be found at <http://www.gnu.org/software/kawa/Software-License.html>.

This product includes OSSP UUID software which is Copyright (c) 2002 Ralf S. Engelschall, Copyright (c) 2002 The OSSP Project Copyright (c) 2002 Cable & Wireless Deutschland. Permissions and limitations regarding this software are subject to terms available at <http://www.opensource.org/licenses/mit-license.php>.

This product includes software developed by Boost (<http://www.boost.org/>) or under the Boost software license. Permissions and limitations regarding this software are subject to terms available at [http://www.boost.org/LICENSE\\_1\\_0.txt](http://www.boost.org/LICENSE_1_0.txt).

This product includes software copyright (c) 1997-2007 University of Cambridge. Permissions and limitations regarding this software are subject to terms available at <http://www.pcre.org/license.txt>.

This product includes software copyright (c) 2007 The Eclipse Foundation. All Rights Reserved. Permissions and limitations regarding this software are subject to terms available at <http://www.eclipse.org/org/documents/epl-v10.php>.

This product includes software licensed under the terms at <http://www.tcl.tk/software/tcltk/license.html>, <http://www.bosrup.com/web/overlib/?License>, <http://www.stlport.org/doc/license.html>, <http://asm.ow2.org/license.html>, <http://www.cryptix.org/LICENSE.TXT>, <http://hsqldb.org/web/hsqLicense.html>, <http://httpunit.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt>, [http://www.gzip.org/zlib/zlib\\_license.html](http://www.gzip.org/zlib/zlib_license.html), <http://www.openldap.org/software/release/license.html>, <http://www.libssh2.org>, <http://slf4j.org/license.html>, <http://www.sente.ch/software/OpenSourceLicense.html>, <http://fusesource.com/downloads/license-agreements/fuse-message-broker-v-5-3-license-agreement>;

<http://antlr.org/license.html>; <http://aopalliance.sourceforge.net/>;  
<http://www.bouncycastle.org/licence.html>; <http://www.jgraph.com/jgraphdownload.html>;  
<http://www.jcraft.com/jsch/LICENSE.txt>; [http://jotm.objectweb.org/bsd\\_license.html](http://jotm.objectweb.org/bsd_license.html); .  
<http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231>;  
<http://www.slf4j.org/license.html>; <http://nanoxml.sourceforge.net/orig/copyright.html>;  
<http://www.json.org/license.html>; <http://forge.ow2.org/projects/javaservice/>,  
<http://www.postgresql.org/about/licence.html>, <http://www.sqlite.org/copyright.html>,  
<http://www.tcl.tk/software/tcltk/license.html>, <http://www.jaxen.org/faq.html>,  
<http://www.jdom.org/docs/faq.html>, <http://www.slf4j.org/license.html>;  
<http://www.iodbc.org/dataspace/iodbc/wiki/iODBC/License>;  
<http://www.keplerproject.org/md5/license.html>; <http://www.toedter.com/en/jcalendar/license.html>;  
<http://www.edankert.com/bounce/index.html>; <http://www.net-snmp.org/about/license.html>;  
<http://www.openmdx.org/#FAQ>; [http://www.php.net/license/3\\_01.txt](http://www.php.net/license/3_01.txt); <http://srp.stanford.edu/license.txt>;  
<http://www.schneier.com/blowfish.html>; <http://www.jmock.org/license.html>; <http://xsom.java.net>; and  
<http://benalman.com/about/license/>;  
<https://github.com/CreateJS/EaselJS/blob/master/src/easeljs/display/Bitmap.js>;  
<http://www.h2database.com/html/license.html#summary>; and <http://jsoncpp.sourceforge.net/LICENSE>.

This product includes software licensed under the Academic Free License  
<http://www.opensource.org/licenses/afl-3.0.php>), the Common Development and Distribution License  
<http://www.opensource.org/licenses/cddl1.php>) the Common Public License  
<http://www.opensource.org/licenses/cpl1.0.php>), the Sun Binary Code License Agreement  
 Supplemental License Terms, the BSD License (<http://www.opensource.org/licenses/bsd-license.php>)  
 the MIT License (<http://www.opensource.org/licenses/mit-license.php>) and the Artistic License  
<http://www.opensource.org/licenses/artistic-license-1.0>).

This product includes software copyright (c) 2003-2006 Joe Walnes, 2006-2007 XStream Committers.  
 All rights reserved. Permissions and limitations regarding this software are subject to terms available  
 at <http://xstream.codehaus.org/license.html>. This product includes software developed by the Indiana  
 University Extreme! Lab. For further information please visit <http://www.extreme.indiana.edu/>.  
 This Software is protected by U.S. Patent Numbers 5,794,246; 6,014,670; 6,016,501; 6,029,178;  
 6,032,158; 6,035,307; 6,044,374; 6,092,086; 6,208,990; 6,339,775; 6,640,226; 6,789,096; 6,820,077;  
 6,823,373; 6,850,947; 6,895,471; 7,117,215; 7,162,643; 7,243,110; 7,254,590; 7,281,001; 7,421,458;  
 7,496,588; 7,523,121; 7,584,422; 7,676,516; 7,720,842; 7,721,270; and 7,774,791, international Patents and  
 other Patents Pending.

**DISCLAIMER:** Informatica Corporation provides this documentation "as is" without warranty of any  
 kind, either express or implied, including, but not limited to, the implied warranties of noninfringement,  
 merchantability, or use for a particular purpose. Informatica Corporation does not warrant that this  
 software or documentation is error free. The information provided in this software or documentation  
 may include technical inaccuracies or typographical errors. The information in this software and  
 documentation is subject to change at any time without notice.

**NOTICES**

This Informatica product (the "Software") includes certain drivers (the "DataDirect Drivers") from  
 DataDirect Technologies, an operating company of Progress Software Corporation ("DataDirect")  
 which are subject to the following terms and conditions:

1. THE DATADIRECT DRIVERS ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER  
 EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF  
 MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.
2. IN NO EVENT WILL DATADIRECT OR ITS THIRD PARTY SUPPLIERS BE LIABLE TO THE END-USER  
 CUSTOMER FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL OR OTHER  
 DAMAGES ARISING OUT OF THE USE OF THE ODBC DRIVERS, WHETHER OR NOT INFORMED OF  
 THE POSSIBILITIES OF DAMAGES IN ADVANCE. THESE LIMITATIONS APPLY TO ALL CAUSES OF  
 ACTION, INCLUDING, WITHOUT LIMITATION, BREACH OF CONTRACT, BREACH OF WARRANTY,  
 NEGLIGENCE, STRICT LIABILITY, MISREPRESENTATION AND OTHER TORTS.

# Table of Contents

1. Introduction.....	4
2. Release UMS 5.3.6 / UMP 5.3.6 / UMQ 5.3.6 - March - 2014 .....	5
3. Release UMS 5.3.5 / UMP 5.3.5 / UMQ 5.3.5 - November - 2013 .....	6
4. Release UMS 5.3.4 / UMP 5.3.4 / UMQ 5.3.4 - July - 2013.....	8
5. Release UMS 5.3.3 / UMP 5.3.3 / UMQ 5.3.3 - March - 2013 .....	12
6. Release UMS 5.3.2 / UMP 5.3.2 - November - 2012 .....	17
7. Release UMS 5.3.1 / UMP 5.3.1 / UMQ 5.3.1 - September - 2012.....	19
8. Release UMS 5.3 / UMP 5.3 / UMQ 5.3 - June - 2012.....	26
9. Release UMS 5.2.2 / UMP 5.2.2 / UMQ 5.2.2 - March - 2012 .....	33
10. Release UMS 5.2.1 / UMP 5.2.1 / UMQ 5.2.1 - February 2012 .....	39
11. Release LBM 4.2.11 / UME 3.2.11 / UMQ 2.1.11 - February 2012.....	43
12. Release UMS 5.2 / UMP 5.2 / UMQ 5.2 - December 2011 .....	45
13. Release UMS 5.1.1 / UMP 5.1.1 / UMQ 5.1.1 - November 2011.....	53
14. Release LBM 4.2.10 / UME 3.2.10 / UMQ 2.1.10 - November 2011 .....	55
15. Release UMS 5.1 / UMP 5.1 / UMQ 5.1 - November 2011.....	57
16. Release LBM 4.2.9 / UME 3.2.9 / UMQ 2.1.9 - November 2011 .....	60
17. Release UMS 5.0.1 / UMP 5.0.1 / UMQ 5.0.1 - September 2011.....	62
18. Release UMS 5.0 / UMP 5.0 / UMQ 5.0 - August 2011 .....	65
19. Release LBM 4.2.8 / UME 3.2.8 / UMQ 2.1.8 - July 2011.....	71
20. Release LBM 4.2.7 / UME 3.2.7 / UMQ 2.1.7 - July 2011.....	74
21. Release LBM 4.2.6 / UME 3.2.6 / UMQ 2.1.6 - June 2011.....	76
22. Release LBM 4.2.5 / UME 3.2.5 / UMQ 2.1.5 - June 2011.....	78
23. Release LBM 4.2.4 / UME 3.2.4 / UMQ 2.1.4 - May 2011 .....	81
24. Release LBM 4.2.3 / UME 3.2.3 / UMQ 2.1.3 - April 2011 .....	83
25. Release LBM 4.2.2 / UME 3.2.2 / UMQ 2.1.2 - April 2011.....	86
26. Release LBM 4.2.1 / UME 3.2.1 / UMQ 2.1.1 - April 2011.....	89
27. Release LBM 4.2 / UME 3.2 / UMQ 2.1 - March 2011 .....	92
28. Release LBM 4.1.3 / UME 3.1.3 / UMQ 2.0.1 - November 2010 .....	99
29. Release LBM 4.1.2 / UME 3.1.2 / UMQ 2.0 - November 2010 .....	100
30. Release UME 3.0.2 - November 2010 .....	103
31. Release LBM 4.1.1 / UME 3.1.1 / UMQ 1.1.1 - October 2010.....	103
32. Release LBM 4.1 / UME 3.1 / UMQ 1.1 - August 2010.....	107
33. Release LBM 4.0.1 - June 2010 .....	112
34. Release LBM 3.6.6 / UME 3.0.1 - August 2010.....	112
35. Release LBM 3.6.5 - May 2010 .....	112
36. Release LBM 4.0 - May 2010 .....	113
37. Release LBM 3.6.3 / UME 3.0.1 / UMQ 1.0 - April 2010.....	118
38. Release LBM 3.6.2 / UME 3.0 - February 2010.....	119
39. Release LBM 3.6.1 - February 2010 .....	120
40. Release LBM 3.6 - December 2009.....	122
41. Release UME 2.2.4 - November 2009 .....	125
42. Release LBM 3.5.3 / UME 2.2.3 - November 2009 .....	125
43. Release LBM 3.6.ea2 / UME 3.0ea2 - September 2009.....	126
44. Release LBM 3.5.2 / UME 2.2.2 - August 2009.....	129
45. Release LBM 3.5.1 / UME 2.2.1 - July 2009 .....	130

# 1. Introduction

This document lists significant changes to **Ultra Messaging®**.

## 2. Release UMS 5.3.6 / UMP 5.3.6 / UMQ 5.3.6 - March - 2014

### 2.1. UMS 5.3.6

#### 2.1.1. Bug Fixes

- 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.
- A source sends Late Join responses with an uninitialized header field, which can cause a fatal assertion on the requesting receiver.

#### 2.1.2. Known Issues

- When calling `lbm_src_send()` from the context thread inside a `lbm_timer` callback, a source can freeze when it hits rate limiter limits, requiring an application restart.

### 2.2. UMP 5.3.6

#### 2.2.1. Bug Fixes

- Fatal assertion `[dest]` at line 481 in `../../../../src/lib/lbm/lbmrxtl.c` in a UM application or daemon if the application receives malformed data on a UM tcp socket or under other extreme edge case conditions.

### 2.3. UMQ 5.3.6

#### 2.3.1. Known Issues

- Configurations with multiple queue instances (slaves) can lead to inconsistent states, which can cause message

loss, crashes or an inability to restart without removal of files (also resulting in message loss). For guidance, see Section 2.2.8 of the Ultra Messaging Guide for Persistence and Queuing.

- A queue instance may crash with the following log message if `allow-queue-browsing` is set to 1:  
[EMERGENCY]: FATAL: failed assertion [sinc\_q\_msg\_info\_node!=NULL] at line 5998 in  
../../../../src/stored/umqueue.c.
- UMQ flags reassigned messages upon delivery, but when a queue restarts, UMQ may re-deliver some messages without flagging them.
- Queues may occasionally crash upon restart if the queue option `forwarding-behavior` is set to `store-while-forward`. Set this option to `store-then-forward`.
- A queue instance may crash with the following log message if messages are of 0 length: [EMERGENCY]: FATAL: failed assertion [msg->lbm\_msg!=NULL] at line 3896 in  
../../../../src/stored/umqsinc.c

## 3. Release UMS 5.3.5 / UMP 5.3.5 / UMQ 5.3.5 - November - 2013

### 3.1. UMS 5.3.5

#### 3.1.1. Updates

- The unicast resolver daemon, `lbmrld`, 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).
- When an error occurs on a socket while sending an ACK in LBT-RU, UM now generates an EOS immediately.
- The following error log messages are added to the Operations Guide:
  - Core-5688-3375: unicast resolver %s:%u went inactive.
  - Core-5688-3387: LBMR Extended Type 0x0 incorrect (%s:%d len %d). [%s]. Dropping.
  - CoreApi-5688-3287: could not find open unicast source port in range [%d-%d]
  - CoreApi-5688-3320: lbm\_socket\_recv recv/recvfrom: %s
- In the .NET API, UM now re-uses `LBMChannelInfo`, `UMQMessageId`, and `UMQIndexInfo` objects between different `LBMMessage` objects for delivery, reducing garbage collection overhead.
- The `lbmrld` resolver daemon now uses blocking sends for all outgoing data.

### 3.1.2. Bug Fixes

- In the UM Gateway, single TCP peer connections under Windows now function correctly.
- UM no longer returns -1 from `lbm_context_process_events()` for non-critical errors.
- Corrected a problem with how UM handles connect failures under Windows, which was causing the UM Gateway to not reconnect with other UM Gateway peer portals after a timeout condition.
- Corrected a problem where a fatal assert would occur when a receiver received a message containing a TSNI request. This could happen when a receiver late joined from an older-version source.

### 3.1.3. Known Issues

- When calling `lbm_src_send()` from the context thread inside a `lbm_timer` callback, a source can freeze when it hits rate limiter limits, requiring an application restart.

### 3.1.4. Version Compatibilities

- LBT-IPC, when used on UM 5.3.x, is not compatible with LBT-IPC used on UM 5.2.x or earlier.

## 3.2. UMP 5.3.5

### 3.2.1. Updates

- In the Ultra Messaging Guide for Persistence, the following sections have been changed or added:
  - 2.1.5 Message Stability
- The following error log message is updated in the Operations Guide:
  - Core-5688-542: received ACK for unknown source from %s:%d\n

## 3.3. UMQ 5.3.5

### 3.3.1. Updates

- In the Ultra Messaging Guide for Persistence and Queuing, the following sections have been changed or added:

- 1.2 Queuing
- 1.3 Semantic Responsibilities
- 2.2.2 Message Stability
- 2.2.3 Once-and-Only-Once Delivery
- 2.2.8 Queue Fault Tolerance
- 2.2.9 Indexed Queuing

If you are considering queuing semantics for your application, please contact Informatica.

- In the Ultra Messaging JMS Guide the following sections have been changed:
  - 1. Introduction, known issues.

### 3.3.2. Known Issues

- Configurations with multiple queue instances (slaves) can lead to inconsistent states, which can cause message loss, crashes or an inability to restart without removal of files (also resulting in message loss). For guidance, see Section 2.2.8 of the Ultra Messaging Guide for Persistence and Queuing.
- A queue instance may crash with the following log message if `allow-queue-browsing` is set to 1:  
[EMERGENCY]: FATAL: failed assertion [sinc\_q\_msg\_info\_node!=NULL] at line 5998 in  
../../../../src/stored/umqueue.c.
- UMQ flags reassigned messages upon delivery, but when a queue restarts, UMQ may re-deliver some messages without flagging them.
- Queues may occasionally crash upon restart if the queue option `forwarding-behavior` is set to `store-while-forward`. Set this option to `store-then-forward`.
- A queue instance may crash with the following log message if messages are of 0 length: [EMERGENCY]: FATAL:  
failed assertion [msg->lbn\_msg!=NULL] at line 3896 in  
../../../../src/stored/umqsinc.c

## 4. Release UMS 5.3.4 / UMP 5.3.4 / UMQ 5.3.4 - July - 2013

### 4.1. UMS 5.3.4

#### 4.1.1. Updates

- The log message, `Core-5688-696: received request for unknown source`, now displays an IP and port



- number. Also, the new message, `Core-7427-1: received TSNI request for unknown source - ip:port[%s:%d]`, differentiates between a request and a TSNI request from an unknown source.
- The following new warning messages indicate when a UM receiver delivery controller reaches its maximum map entries, which could cause Late Join to end prematurely.
    - `CoreApi-7506-1: rdelvc force loss due to dctrl_entries exceeding ctx_max_entries!`
    - `CoreApi-7506-2: rdelvc force loss when doing Late Join or OTR due to dctrl_entries exceeding ctx_max_entries!`
  - The log message `Core-5688-7 FATAL: failed assertion...` has been removed. In addition to removing this message, a backtrace has been added to platforms with backtrace support.
  - UM now uses the multicast interface in the OTID to reduce the possibility of an OTID collision in sources. A collision of a source OTID can cause a receiver to join only one of the source streams as it sees the second as a duplicate.
  - The `resolver_unicast_daemon` option now issues the new log message, `CoreApi-7875-1: optval cannot contain multiple values`, to indicate that multiple values are not allowed for this option:
  - The `lbmsdm.h` file now contains a check to not typedef `intx_t` types for Windows platforms that have these already defined.

#### 4.1.2. Bug Fixes

- Corrected a problem that caused `lbmrd` to propagate non-LBMR messages to clients.
- UM now allows for NAT translation of request/response and Late Join.
- Corrected a problem that could cause indefinite `EWOULDBLOCK` on the Windows platform.
- Corrected a problem where an active `lbmrd` erroneously logged message `LOG Level 6: Core-5688-3375: unicast resolver xxx.xxx.xxx.xxx:xxxx went inactive`

#### 4.1.3. Known Issues

- The .NET API ([../DotNetAPI/doc/Index.html](http://../DotNetAPI/doc/Index.html)) does not support message selectors with unicode symbols and causes receiver creation to fail.
- UM does not support the use of Unicode PCRE characters in wildcard receiver patterns on any system that communicates with a HP-UX or AIX system.
- Hot Failover is not currently supported with Multitransport Threads.
- Configuring sources with LBT-RU, source-side filtering, and adaptive batching may cause a crash. All three options must be set to cause the problem. Informatica is working to address this issue.
- When using the UM Gateway Informatica recommends that you **do not** enable the Multitransport Threads feature. Informatica does not support nor test the operation of Multitransport Threads across the UM Gateway.
- Multitransport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), UMS 4.1 issues the following warning: LOG Level 5:  
LBT-RDMA: VMS Logger Message (Error): vmss\_create\_store: 196[E] vms\_listen:  
rdma\_bind\_addr failed (r=-1). This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, UMS may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support gateway failover, MIM, persistence, queuing, JMS, or Ultra Load Balancing. Gateway support of these features is in development.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).
- If proxy sources are enabled and gateways bounce, you may see a fatal assertion.

#### 4.1.4. Version Compatibilities

- Although UMS 5.0 applications should run well with LBM 4.2 UM Gateways, all Gateways should always run the same version.

## 4.2. UMP 5.3.4

### 4.2.1. Updates

- Port=0 is no longer a valid port number for umestored (UMP or UMQ). Log Messages `Store-5688-5408` and `Store-5688-5410` now indicate that port 0 is no longer valid. The valid port range in the messages is now `[1,65535]`.
- A webmon port in the umestored configuration file now must be greater than 0. Specifying a port of 0 or anything that would be translated to a value less than 0 prohibits umestored from starting and instead you see `web-monitor port must be greater than 0.`
- Fragments requested for retransmission by a receiver from a store may now be declared as unrecoverable prior to the retransmission request timeout, if all stores have declared that fragment unrecoverable.
- Log messages `Store-5688-5332` and `Store-5688-5333` are added to the Operations Guide.

### 4.2.2. Bug Fixes

No problems specific to **UMP** were addressed.

### 4.2.3. Known Issues

- Creating multiple receivers on the same topic in the same context using the Java API (`../JavaAPI/html/index.html`) may result in a crash when calling `Dispose()` if the topic is a persistent data stream. It is recommended to either protect the calls to `Dispose()` to prevent concurrent calls or use a single receiver per topic and use application code to deliver messages to different callbacks. This will be fixed in a future release.
- **UMP** stores sometimes fail to send message stability acknowledgements to sources for messages sent while the store was being reported unresponsive by the source. Informatica is investigating this problem.
- A **UMP** store daemon, `umestored`, may crash if you enable proxy sources and the daemon is configured with a very restrictive port range, (`transport_lbtrm_multicast_address_low` (`../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTLOW`) - `transport_lbtrm_multicast_address_high` (`../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTHIGH`), `transport_tcp_port_low` (`../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPPORTLOW`) - `transport_tcp_port_high` (`../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPPORTHIGH`), etc.). Informatica recommends that you use a wide port range (at least 1000 ports) in your **UM** configuration file if you enable proxy sources for `umestored`. Informatica is investigating this problem.
- 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. As a workaround, Informatica recommends that you enable `receiver_delivery_control_loss_check_interval` (`../Config/deliverycontroloptions.html#RECEIVERDELIVERYCONTROLLOSSCHECKINTERVAL`) 2500 in the **UM** configuration file, not the `umestored` XML configuration file. The value of 2500 assumes the default

- \*\_nak\_generation\_interval. See Preventing Undetected Loss ([../Config/grouprelated.html#PREVENTUNDETECTEDLOSS2](#)) for more information. Informatica is investigating this problem.
- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use LBM\_EVQ\_BLOCK when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. Informatica is working on a solution to this problem.
  - 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]. Changing `fd_management_type` ([../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE](#)) to `devpoll` prevents this problem. Informatica is investigating this problem.
  - **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 4.3. UMQ 5.3.4

### 4.3.1. Bug Fixes

- Corrected a problem that allowed ULB to reassign messages even if `ump_ulb_application_set_message_max_reassignments` was set to 1. The user document correctly states that by setting the flag to 1, a message will not be reassigned.

### 4.3.2. Known Issues

- The method, `getJMSRedelivered` for a message always returns false for redelivered messages when using Queue Session IDs along with the topic's `message-reassignment-timeout` set to 0 (zero).
- The UM configuration file specified for a queue cannot also contain source attributes for a store such as, `source_ume_store 127.0.0.1:14567` or `source_ume_store_name NYstore2`.
- When servicing large message retrieval requests from a queue browser, the queue may become busy enough that the receiver perceives it as unresponsive and re-registers with the queue, canceling the message retrieval request. Informatica recommends limiting large message retrievals to 1000 messages at a time, waiting for each retrieval to finish before beginning the next retrieval request

## 5. Release UMS 5.3.3 / UMP 5.3.3 / UMQ 5.3.3 - March - 2013

### 5.1. UMS 5.3.3

#### 5.1.1. Updates

- **UM** no longer sends a Topic Management Record (TMR) when an application deletes a receiver in a standard way with, for example `lbm_rcv_delete()`, when no UM Gateways are in use. TMRs specifically inform a UM Gateway that a receiver for a particular topic has been shutdown. Without the presence of UM Gateways, TMRs become unnecessary network traffic.
- The following error log messages have been added to the Concepts Guide section, UM Log Messages (`./Design/logmessages.html`).
  - Core-7421-1: Source Side Filtering Init message with no return IP, using transport IP (`%s`).
  - Core-7479-1: NOTICE from src (RID:%u: (`%s`)): store `%s:%u` reports it has not received TIR. Possible misconfiguration?
  - CoreAPI-6783-1: `lbm_socket_sendb` send error occurred while sending. The message will contain addition specific information, supplied by the operating system.
  - Gwd-6033-353: : endpoint portal [`%s`] received one or more UIM control messages with no stream information - these will be dropped.
  - Gwd-6033-593: peer portal [`%s`] received one or more UIM control messages with no stream information - these will be dropped.

#### 5.1.2. Bug Fixes

- Corrected a problem with **UM** sockets that may cause a crash while binding TCP ports.
- Fixed a problem with `lbm_set_umm_info()` 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 logon error or a communications error with the UMM Daemon.
- 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. Configuration files over 1K will result in an error: error reading config file. Informatica will fix this in a future release. A workaround is to convert the plain text configuration file to an XML configuration file.*

### 5.1.3. Known Issues

- The .NET API (./DotNetAPI/doc/Index.html) does not support message selectors with unicode symbols and causes receiver creation to fail.
- **UM** does not support the use of Unicode PCRE characters in wildcard receiver patterns on any system that communicates with a HP-UX or AIX system.
- Hot Failover is not currently supported with Multitransport Threads.
- Configuring sources with LBT-RU, source-side filtering, and adaptive batching may cause a crash. All three options must be set to cause the problem. Informatica is working to address this issue.
- When using the UM Gateway Informatica recommends that you **do not** enable the Multitransport Threads feature. Informatica does not support nor test the operation of Multitransport Threads across the UM Gateway.
- Multitransport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (./Config/majoroptions.html#RECEIVERORDEREDEDELIVERY) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (./Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): vmss\_create\_store: 196[E] vms\_listen: rdma\_bind\_addr failed (r=-1). This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (./Config/automaticmonitoringoptions.html) with `ud_acceleration` (./Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION) and the `epoll` file descriptor option, **UMS** may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support gateway failover, MIM, persistence, queuing, JMS, or Ultra Load Balancing. Gateway support of these features is in development.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.

- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

#### 5.1.4. Version Compatibilities

- Although **UMS 5.0** applications should run well with **LBM 4.2** UM Gateways, all Gateways should always run the same version.

## 5.2. UMP 5.3.3

### 5.2.1. Updates

- The following error log messages have been added to the Operations Guide.
  - Core-7421-1: Source Side Filtering Init message with no return IP, using transport IP (%s).
  - Core-7479-1: NOTICE from src (RID:%u: (%s)): store %s:%u reports it has not received TIR. Possible misconfiguration?
  - CoreAPI-6783-1: `lbm_socket_sendb` send error occurred while sending. The message will contain addition specific information, supplied by the operating system.
  - Gwd-6033-353: : endpoint portal [%s] received one or more UIM control messages with no stream information - these will be dropped.
  - Gwd-6033-593: peer portal [%s] received one or more UIM control messages with no stream information - these will be dropped.
- 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: `UME registration error: error in UME registration, RegID 1508542809 is in use by a different receiver at store 0. Previously the message always displayed RegID 0 and no store number.`
- 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.`

## 5.2.2. Bug Fixes

- 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.
- Corrected an infrequent problem that could cause a store (`umestored`) to perpetually send TSNI requests to a **UM** source at the `retransmit_request_interval` (`../Config/latejoinoptions.html#RECEIVERRETRANSMITREQUESTINTERVAL`) after the **UM** source has resumed sending messages.
- 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.
- Corrected an issue where stability acknowledgements from a persistent store to the **UM** source during a store proxy source election could be sent with invalid topic and transport indexes, which the source would discard.

## 5.2.3. Known Issues

- Creating multiple receivers on the same topic in the same context using the Java API (`../JavaAPI/html/index.html`) may result in a crash when calling `Dispose()` if the topic is a persistent data stream. It is recommended to either protect the calls to `Dispose()` to prevent concurrent calls or use a single receiver per topic and use application code to deliver messages to different callbacks. This will be fixed in a future release.
- **UMP** stores sometimes fail to send message stability acknowledgements to sources for messages sent while the store was being reported unresponsive by the source. Informatica is investigating this problem.
- A **UMP** store daemon, `umestored`, may crash if you enable proxy sources and the daemon is configured with a very restrictive port range, (`transport_lbtrm_multicast_address_low` (`../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTLOW`) - `transport_lbtrm_multicast_address_high` (`../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTHIGH`), `transport_tcp_port_low` (`../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPPORTLOW`) - `transport_tcp_port_high` (`../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPPORTHIGH`), etc.). Informatica recommends that you use a wide port range (at least 1000 ports) in your **UM** configuration file if you enable proxy sources for `umestored`. Informatica is investigating this problem.
- 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. As a workaround, Informatica recommends that you enable `receiver_delivery_control_loss_check_interval` (`../Config/deliverycontroloptions.html#RECEIVERDELIVERYCONTROLLOSSCHECKINTERVAL`) 2500 in the **UM** configuration file, not the `umestored` XML configuration file. The value of 2500 assumes the default `*_nak_generation_interval`. See `Preventing Undetected Loss` (`../Config/grouprelated.html#PREVENTUNDETECTEDLOSS2`) for more information. Informatica is investigating this problem.



- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. Informatica is working on a solution to this problem.
- 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]`. Changing `fd_management_type` (`./Config/majoroptions.html#CONTEXTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 5.3. UMQ 5.3.3

### 5.3.1. Bug Fixes

- Corrected an infrequent condition that caused the fatal assert `[prop_offset >= 0]` if a **UMQ** receiver received a message containing message properties.
- Corrected an issue that prevented a **UMQ** receiver for a dead letter topic from registering with a queue if a particular queue instance did not have a record of the creation of the dead letter topic.
- Corrected a problem that caused a segmentation fault when restarting a queue if the queue daemon (`umestored`) was configured with a disk sync log file (`sync-log-filename`).

### 5.3.2. Known Issues

- The method, `getJMSRedelivered` for a message always returns `false` for redelivered messages when using Queue Session IDs along with the topic's `message-reassignment-timeout` set to 0 (zero).
- The **UM** configuration file specified for a queue cannot also contain source attributes for a store such as, `source_ume_store 127.0.0.1:14567` or `source_ume_store_name NYstore2`.
- When servicing large message retrieval requests from a queue browser, the queue may become busy enough that the receiver perceives it as unresponsive and re-registers with the queue, canceling the message retrieval request. Informatica recommends limiting large message retrievals to 1000 messages at a time, waiting for each retrieval to finish before beginning the next retrieval request

## 6. Release UMS 5.3.2 / UMP 5.3.2 - November - 2012

### 6.1. UMS 5.3.2

No changes have been made to **UMS** for 5.3.2. This release accompanies the **UMP** 5.3.2 release which provides support for the HP NonStop® platform.

### 6.2. UMP 5.3.2

#### 6.2.1. New

- **Ultra Messaging Persistence Edition** is now available on the HP NonStop platform. See HP NonStop® Package (../DocIntro.html#HPNONSTOPPACKAGES) for limitations.

#### 6.2.2. Bug Fixes

No specific **UMP** problems were discovered.

#### 6.2.3. Known Issues

- Creating multiple receivers on the same topic in the same context using the Java API (../JavaAPI/html/index.html) may result in a crash when calling `Dispose()` if the topic is a persistent data stream. It is recommended to either protect the calls to `Dispose()` to prevent concurrent calls or use a single receiver per topic and use application code to deliver messages to different callbacks. This will be fixed in a future release.
- **UMP** stores sometimes fail to send message stability acknowledgements to sources for messages sent while the store was being reported unresponsive by the source. Informatica is investigating this problem.
- A **UMP** store daemon, `umestored`, may crash if you enable proxy sources and the daemon is configured with a very restrictive port range, (`transport_lbtrm_multicast_address_low` (../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTLOW) - `transport_lbtrm_multicast_address_high` (../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTHIGH), `transport_tcp_port_low` (../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPPORTLOW) - `transport_tcp_port_high` (../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPPORTHIGH), etc.). Informatica recommends that you use a wide port range (at least 1000 ports) in your **UM** configuration file if you enable proxy sources for `umestored`. Informatica is investigating this problem.
- 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. As a workaround, Informatica recommends that you enable `receiver_delivery_control_loss_check_interval` (../Config/deliverycontroloptions.html#RECEIVERDELIVERYCONTROLLOSSCHECKINTERVAL) 2500 in

the **UM** configuration file, not the `umestored` XML configuration file. The value of 2500 assumes the default `*_nak_generation_interval`. See [Preventing Undetected Loss \(../Config/grouprelated.html#PREVENTUNDETECTEDLOSS2\)](#) for more information. Informatica is investigating this problem.

- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. Informatica is working on a solution to this problem.
- 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]`. Changing `fd_management_type` ([../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE](#)) to `devpoll` prevents this problem. Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 7. Release UMS 5.3.1 / UMP 5.3.1 / UMQ 5.3.1 - September - 2012

### 7.1. UMS 5.3.1

#### 7.1.1. New Features

- **Ultra Messaging Manager** has been added to **UMS**. Previously this feature was only available with **UMP** and **UMQ**. This feature ensures the consistency and reliability of enterprise production applications by enabling IT administrators to control what configurations messaging applications can use and what users can operate them. See [Ultra Messaging Manager \(../UMM/index.html\)](#).

#### 7.1.2. Updates

- The **UM** log message, `Core-5688-1866`, has been changed from Warning to an INFO message.
- Performance has been improved by converting locks for `srcs` and `imbq` to spinlocks in the Windows code. The environment variable `LBM_WINDOWS_RATELIMITER_SPINLOCK_COUNT` sets the spinlock count. `InitializeCriticalSection` will be used if the variable is not set. `InitializeCriticalSectionAndSpinCount` will be used when the variable is set.

### 7.1.3. Bug Fixes

- Fixed a problem in the .NET API (../DotNetAPI/doc/Index.html) that caused an exception when assigning a UNICODE string to the `LBMReceiver` topic attribute.
- Fixed an issue on AIX platforms that resulted in the message selector string `unknown` to be parsed incorrectly as `true`.
- Corrected a problem in PDM that caused out of bounds exceptions when calling `setValue(PDMMessage [])` after modifying a PDM message that is part of a `MESSAGE_ARR` field within another serialized PDM message. After modifying a PDM message that is member of a `MESSAGE_ARR` type field, You must call `setValue(PDMMessage [])` to allow PDM to update the internal version of the message.
- Changed the `fd_management_type` (../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE) to `epoll` so that it no longer asserts when closing STDIN.
- Corrected a problem with multiple receivers for a topic in adjacent Topic Resolution Domains serviced by a UM Gateway that caused topic interest to not be properly represented in the adjacent domains. This resulted in remote receivers not getting the expected message data.
- Fixed a problem with Message Selectors which could result in a failure when comparing NaN values.
- Fixed a problem that caused receivers to get corrupted packets when the source does a non-blocking send over TCP without an event queue.
- Fixed a problem that could result in the error log message, `CoreApi-5688-3233: TCP server listen: (98)` even if TCP ports are still available.
- Fixed an issue with PDM that caused a segfault when using different message versions and a different number of fields. This correction was applied to the C API (../API/index.html), the Java API (../JavaAPI/html/index.html) and the .NET API (../DotNetAPI/doc/Index.html).
- Reversed a UM 5.3 bug fix because it was shown to produce receiver deafness across a UM Gateway. The UM 5.3 bug fix change log entry appears below.

*Added additional validation logic to Topic Resolution advertisement (TIR) processing to prevent malformed packets from causing an invalid memory access, potentially leading to a crash.*

- Corrected a problem that now allows Late Join to work properly when using the `network_compatibility_mode` (../Config/majoroptions.html#CONTEXTNETWORKCOMPATIBILITYMODE).
- Fixed an issue that caused a JVM crash if multiple contexts were created with the same port.
- Corrected problems with uninitialized attribute values for context, source, receiver, and event queue attributes.
- Fixed a problem with sending and receiving message properties between platforms using different endian formats.
- JMS now creates its own MessageID rather than using one based on the underlying JMS system. Also added are UM-layer functions that create a `JMSMessageId`, to add compatibility between JMS and the other UM senders/receivers.
- Fixed a problem that exposed LBMC to possible Denial of Service (DoS) attacks.
- Fixed a problem that caused a fatal assert `Core-5688-7: FATAL: failed assertion` that could happen when setting `network_compatibility_mode`.

- Fixed another problem that sometimes caused fatal assert `Core-5688-7: FATAL: failed assertion`. **UM** now creates a unique mutex name that results in successful creation of a named **Mutex**, thus avoiding the assert.

#### 7.1.4. Known Issues

- Retrieving plain text (not XML) configurations files via an http or ftp request is limited to configuration files of 1K in size. Configuration files over 1K will result in an error: `error reading config file`. Informatica will fix this in a future release. A workaround is to convert the plain text configuration file to an XML configuration file.
- The .NET API (`../DotNetAPI/doc/Index.html`) does not support message selectors with unicode symbols and causes receiver creation to fail.
- **UM** does not support the use of Unicode PCRE characters in wildcard receiver patterns on any system that communicates with a HP-UX or AIX system.
- Hot Failover is not currently supported with Multitransport Threads.
- Configuring sources with LBT-RU, source-side filtering, and adaptive batching may cause a crash. All three options must be set to cause the problem. Informatica is working to address this issue.
- When using the UM Gateway Informatica recommends that you **do not** enable the Multitransport Threads feature. Informatica does not support nor test the operation of Multitransport Threads across the UM Gateway.
- Multitransport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: `LOG Level 5: LBT-RDMA: VMS Logger Message (Error): vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using `Automatic Monitoring` (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **UMS** may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support gateway failover, MIM, persistence, queuing, JMS, or Ultra Load Balancing. Gateway support of these features is in development.

- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

### 7.1.5. Version Compatibilities

- Although **UMS 5.0** applications should run well with **LBM 4.2** UM Gateways, all Gateways should always run the same version.

## 7.2. UMP 5.3.1

### 7.2.1. Updates

- The notice Core-5688-3847 can be received when setting the `fd_management_type` (`./Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) to `wincompport` only. The occurrence of this notice has been changed so that after the first 5 occurrences, the notice only prints every 10,000th consecutive occurrence. An example of this notice: `Core-5688-3847: NOTICE: wincport 00000000893BE90 line 4615 WSA err 10061, Connection refused (peer 10.29.3.56:23005) (op 4)`
- Added identifying information such as RegID, store name and store index to registration error responses.
- The UM persistent store has been modified to retain state and cache files when the store restarts that may be corrupted instead of deleting them. The store renames these files to `regid-state/cache.pid.month_day_year`. These files remain indefinitely in the same cache/state directories as specified in the store's XML configuration file.

### 7.2.2. Bug Fixes

- Corrected a problem that prevented a restarted store from correctly persisting the receiver lifetime timeouts. NOTE: Be sure to delete the old state files when upgrading your system to this release.
- Corrected a problem where a restarted unemstored process generated unneeded log messages upon encountering empty cache files.

- Corrected a problem that caused a memory store to attempt to late-join messages from the source that the repository has already received. This occurred after the source re-registers with the store. Now the repository only initiates Late Join for messages that it has not received.
- Corrected a potential memory corruption issue in the store daemon (`umestored`), that was introduced in UM 5.3. The memory corruption could occur when the store, upon startup, detects and deletes a corrupt or empty cache file.
- Corrected a condition where a **UMP** store configured with Receiver-paced Persistence and a `repository-type` of `disk` could stop persisting new data if messages at the end of the cache file were not consumed by all registered receivers and there was available space at the beginning of the cache file.
- Corrected a problem with receivers registering with Quorum/Consensus stores that resulted in the log message:  

```
Core-5688-3113: NOTICE: UME receiver "name" source
"LBTRM:10.29.3.16:14390:228383bb:224.10.10.10:14400" registration consensus sequence
number 0 less than highest seen 51c. Using consensus. Receivers now properly register with all
stores in the group.
```
- Fixed an issue that would incorrectly allow multiple receivers on the same topic with the same `ume_session_id` (`../Config/ultramessagingpersistenceoptions.html#RECEIVERUMESESSSIONID`) to both be registered with the store simultaneously.
- Fixed a problem that caused a large delay in receiver re-registration if a source goes down and the store creates a proxy source or if the source simply restarts.
- Fixed the store daemon (`umestoreds`) shutdown issue on Microsoft® Windows® so that the store does not log an error message when the store shutdown command is issued via the Service Control Manager (SCM).
- Corrected a problem with store initialization that could cause a segfault during long running store recoveries.
- Corrected a problem with restarted stores that produced the error log message, LOG Level 5:  

```
Core-5688-3167: WARNING: received keepalive from non-active store 0. Now a store does not
send keepalives to a source if the source is not registered to the store.
```
- Corrected a problem where a receiver on the opposite side of a gateway did not receive a persistent message until a persistent source in the same context sent one.

### 7.2.3. Known Issues

- Creating multiple receivers on the same topic in the same context using the Java API (`../JavaAPI/html/index.html`) may result in a crash when calling `Dispose()` if the topic is a persistent data stream. It is recommended to either protect the calls to `Dispose()` to prevent concurrent calls or use a single receiver per topic and use application code to deliver messages to different callbacks. This will be fixed in a future release.
- **UMP** stores sometimes fail to send message stability acknowledgements to sources for messages sent while the store was being reported unresponsive by the source. Informatica is investigating this problem.
- A **UMP** store daemon, `umestored`, may crash if you enable proxy sources and the daemon is configured with a very restrictive port range, (`transport_lbtrm_multicast_address_low` (`../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTLOW`) - `transport_lbtrm_multicast_address_high` (`../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTHIGH`), `transport_tcp_port_low` (`../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPPORTLOW`) -

`transport_tcp_port_high`

(`../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPPTHIGH`), etc.). Informatica recommends that you use a wide port range (at least 1000 ports) in your **UM** configuration file if you enable proxy sources for `umestored`. Informatica is investigating this problem.

- 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. As a workaround, Informatica recommends that you enable `receiver_delivery_control_loss_check_interval` (`../Config/deliverycontroloptions.html#RECEIVERDELIVERYCONTROLLOSSCHECKINTERVAL`) 2500 in the **UM** configuration file, not the `umestored` XML configuration file. The value of 2500 assumes the default `*_nak_generation_interval`. See [Preventing Undetected Loss](#) (`../Config/grouprelated.html#PREVENTUNDETECTEDLOSS2`) for more information. Informatica is investigating this problem.
- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. Informatica is working on a solution to this problem.
- 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]`. Changing `fd_management_type` (`../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 7.3. UMQ 5.3.1

### 7.3.1. Updates

- Added `umq_command_outstanding_maximum` (`../Config/ultramessagingqueuingoptions.html#CONTEXTUMQCOMMANDOUTSTANDINGMAXIMUM`) to control the maximum number of outstanding **UMQ** commands (registrations, de-registrations, indexed queuing commands, message list commands, etc.) a client is allowed at one time. Outstanding commands are not yet acknowledged by the queue.
- Updated the JMS example, `Reply.java`, to fix a case where it would fail to reply to a native source that did not correctly set the `JMSReplyTo` property. Also, updated an internal exception to more cleanly handle the case where an application attempts to send to a null destination.

### 7.3.2. Bug Fixes

- Fixed a problem that could cause a persistent **UMQ** queue daemon (`umestored`) to occasionally fail to start up if



it previously had registered receivers that used **UMQ** session IDs.

- Fixed an issue that could cause a non-**UMQ** context receiving a unicast **UMQ** control message to crash. This only happened if a non-**UMQ** context was created and re-used a request port from a previously running **UMQ** context.
- Fixed a small memory leak during the clean shutdown of a slave instance of the **UMQ** queue daemon (`umestored`). This leak had no impact during normal operation.
- Fixed a problem that could cause **UMQ** clients to crash if they had over 65535 currently outstanding queue commands. This many queue commands could accumulate if you performed a very large message retrieve command or attempted to register tens of thousands of clients with a queue in a very short period of time.
- Fixed a problem that could cause a persistent, disk queue daemon (`umestored`) to sometimes fail to restart with a `anode!=NULL` assertion.
- Corrected problems with **UMQ** session IDs and observer receivers (`umq_queue_participation` (`../Config/ultramessagingqueuingoptions.html#RECEIVERUMQQUEUEPARTICIPATION`) = 2) that could cause the queue daemon (`umestored`) to crash upon shutdown if multiple observer receivers on the same topic with different session IDs were used.
- Fixed a problem with how Ultra Messaging JMS tracks sources within a context that resulted in sources on a topic being created multiple times.
- Corrected a problem in Ultra Messaging JMS that prevented multiple `MessageProducers` with the same topic on the same connection.
- Corrected a problem that caused a crash when **UMQ** Ultra Load Balancing receivers were assigned incorrect Application IDs during registration.
- Fixed an issue that resulted in received message timestamp values to be set to 0 (zero) for **UMQ** messages.
- Fixed a problem that caused the JMS redelivered flag to not be set on redelivered messages when **UMQ** was in linear mode.
- Added the following new constants to `lbm.h`, `lbm.java` and `lbm.cs` to aid in interoperability with Ultra Messaging JMS.

```
LBM_TEXTMESSAGE = 0;
LBM_BYTEMESAGE = 1;
LBM_MAPMESSAGE = 2;
LBM_MESSAGE = 3;
LBM_OBJECTMESSAGE = 4;
LBM_STREAMMESSAGE = 5;
LBM_JMSDeliveryMode = "JMSDeliveryMode";
LBM_JMSExpiration = "JMSExpiration";
LBM_JMSPriority = "JMSPriority";
LBM_JMSMessageID = "JMSMessageID";
LBM_JMSTimestamp = "JMSTimestamp";
LBM_JMSCorrelationID = "JMSCorrelationID";
LBM_JMSType = "JMSType";
LBM_JMSRedelivered = "JMSRedelivered";
LBM_LBMMessageType = "LBMMessageType";
LBM_JMSTopicType = "JMSTopicType";
LBM_JMSReplyToName = "JMSReplyToName";
LBM_JMSReplyToWildcard = "JMSReplyToWildcard";
LBM_JMSReplyToType = "JMSReplyToType";
```

### 7.3.3. Known Issues

- The method, `getJMSRedelivered` for a message always returns `false` for redelivered messages when using Queue Session IDs along with the topic's `message-reassignment-timeout` set to 0 (zero).
- The UM configuration file specified for a queue cannot also contain source attributes for a store such as, `source_ume_store 127.0.0.1:14567` or `source_ume_store_name NYstore2`.
- When servicing large message retrieval requests from a queue browser, the queue may become busy enough that the receiver perceives it as unresponsive and re-registers with the queue, canceling the message retrieval request. Informatica recommends limiting large message retrievals to 1000 messages at a time, waiting for each retrieval to finish before beginning the next retrieval request

## 8. Release UMS 5.3 / UMP 5.3 / UMQ 5.3 - June - 2012

### 8.1. UMS 5.3

#### 8.1.1. Updates

- The behavior of the context attribute `network_compatibility_mode` (`../Config/majoroptions.html#CONTEXTNETWORKCOMPATIBILITYMODE`) was enhanced to provide compatibility with each UM release from LBM 3.6 to the present. See also Network Compatibility Mode (`../Config/reference.html#NETWORKCOMPATIBILITY`).
- The UM Gateway is now a separate download package.
- Added additional Microsoft redistributable DLL's to UM Microsoft Windows packages to accompany MSVCR80.
- **Off Transport Recovery (OTR)** - Modified this feature to flag messages recovered by OTR with the OTR flag. In the previous version, OTR flagged recovered messages with the RETRANSMIT flag. (This may involve a code change for some users.) Also, OTR now initiates independently of NAK behavior by receivers. Added 3 new OTR configurations options.
  - `otr_request_initial_delay`  
(`../Config/off-transportrecoveryoptions.html#RECEIVEROTRREQUESTINITIALDELAY`)
  - `otr_request_log_alert_cooldown`  
(`../Config/off-transportrecoveryoptions.html#RECEIVEROTRREQUESTLOGALERTCOOLDOWN`)
  - `otr_request_maximum_interval`  
(`../Config/off-transportrecoveryoptions.html#RECEIVEROTRREQUESTMAXIMUMINTERVAL`)
 See also Off-Transport Recovery (OTR) (`../Design/lbm-features.html#OTR`).
- The SmartHeap library has been removed from the UM distribution, as it is not used by UM.

- Changed all **UM** daemons to print consistent version information when using the `-h` option.
- **Important** - Users are strongly advised not to configure any port or port range with values which fall within your platform's ephemeral port range. This includes the `umstored` port, LBT-TCP source ports, resolver UDP ports, RM destination and source ports or any port that you can configure. All **UM** default ports explicitly avoid all platform ephemeral port values.

### 8.1.2. Bug Fixes

- Resolved the Known Issue: *When using the LBT-IPC Resource Manager ([../Design/lbm-objects.html#LBT-IPC-RESOURCE-MANAGER](#)) tool to view the currently allocated IPC transport sessions, the Session ID is presented in the reverse byte order from what is given to the application via the source string.* The byte order and order shared memory region name now present correctly.
- Corrected various memory leaks associated with the Arrival Order with Reassembly feature. (`ordered_delivery` ([../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY](#)) = -1.)
- Corrected a problem with Arrival Order with Reassembly that could cause a crash when reassembling messages with more than 256 fragments. (`ordered_delivery` ([../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY](#)) = -1.)
- Corrected an issue with the UM Gateway prevented a persistent source from resolving the name of a store which had been restarted.
- Added additional validation logic to Topic Resolution advertisement (TIR) processing to prevent malformed packets from causing an invalid memory access, potentially leading to a crash.
- Corrected a problem with the LBT-IPC transport that caused a **UM** source to abort when multiple receivers were killed while the source was sending data. A lock was added to serialize receiver deletion.
- Fixed an issue in the C API ([../API/index.html](#)) only that could cause a SEGV if a you reuse a message properties object.
- Fixed an issue in the C API ([../API/index.html](#)) only where properties could be duplicated if you reused the same message properties object.
- Fixed a small memory leak when creating multiple receivers on the same topic and context.
- Corrected a problem with LBT-IPC on Microsoft Windows that caused receivers to stop receiving messages.
- Corrected the Java API ([../JavaAPI/html/index.html](#)) `isFragment()` method to return TRUE if a message is a Fragment and false if it is an assembled message, just like the .NET API ([../DotNetAPI/doc/Index.html](#)). Also created a new C API ([../API/index.html](#)) function, `lbm_msg_is_fragment()`.

### 8.1.3. Known Issues

- The .NET API ([../DotNetAPI/doc/Index.html](#)) does not support message selectors with unicode symbols and causes receiver creation to fail.
- **UM** does not support the use of Unicode PCRE characters in wildcard receiver patterns on any system that communicates with a HP-UX or AIX system.
- Hot Failover is not currently supported with Multitransport Threads.

- Configuring sources with LBT-RU, source-side filtering, and adaptive batching may cause a crash. All three options must be set to cause the problem. Informatica is working to address this issue.
- When using the UM Gateway Informatica recommends that you **do not** enable the Multitransport Threads feature. Informatica does not support nor test the operation of Multitransport Threads across the UM Gateway.
- Multitransport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`./Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`./Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`./Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`./Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **UMS** may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support gateway failover, MIM, persistence, queuing, JMS, or Ultra Load Balancing. Gateway support of these features is in development.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

## 8.1.4. Version Compatibilities

- Although **UMS** 5.0 applications should run well with **LBM** 4.2 UM Gateways, all Gateways should always run the same version.

## 8.2. UMP 5.3

### 8.2.1. New Features

- **Receiver-paced Persistence (RPP)** - This feature retains RPP messages until all RPP receivers acknowledge consumption. The repository maintains an accurate count of all RPP receivers. You enable RPP with **UM** configuration options. No special API calls are needed. See Receiver-paced Persistence Operations ([../UME/ume-ops-operational-view.html#UMP-RPPOPS](#)).
- Added the repository type `reduced-fd`, which reduces the number of file descriptors used by **UM**. See Source Repositories ([../UME/architectures.html#SOURCE-REPOSITORIES](#)).
- Added the **UM** configuration option, `ume_flight_size_bytes` ([../Config/ultramessagingpersistenceoptions.html#SOURCEUMEFLIGHTSIZEBYTES](#)).
- Enabled **UM** to send retransmissions from a thread separate from the main context thread so as not to impede live message data processing. The `<store>` configuration option, `retransmission-request-processing-rate`, sets the store's capacity to process retransmission requests. See also Options for a Store's `ume-attributes` Element ([../UME/umestored-config.html#UMESTORED-STORE-OPTIONS](#)) and Receiver-paced Persistence Operations ([../UME/ume-ops-operational-view.html#UMP-RPPOPS](#)).
- Added the **UM** configuration option, `ume_repository_ack_on_reception` ([../Config/ultramessagingpersistenceoptions.html#SOURCEUMEREPOSITORYACKONRECEPTION](#)). When set by a **UM** source in a RPP configuration, this option's value can reset the repository's option, `repository-allow-ack-on-reception`. See also Options for a Topic's `ume-attributes` Element ([../UME/umestored-config.html#UMESTORED-TOPIC-OPTIONS](#)) and Receiver-paced Persistence Operations ([../UME/ume-ops-operational-view.html#UMP-RPPOPS](#)).
- Added the **UM** configuration option, `ume_write_delay` ([../Config/ultramessagingpersistenceoptions.html#SOURCEUMEWRTTEDELAY](#)). When set by a **UM** source in a RPP configuration, this option's value can reset the repository's option, `repository-disk-write-delay`. See also Options for a Topic's `ume-attributes` Element ([../UME/umestored-config.html#UMESTORED-TOPIC-OPTIONS](#)) and Receiver-paced Persistence Operations ([../UME/ume-ops-operational-view.html#UMP-RPPOPS](#)).

### 8.2.2. Updates

- **UMP** store daemons (`umestored`) now include a hexadecimal session ID in log messages that include a registration ID. Also all registration ID are now printed as an unsigned decimal number.

- Added improvements to speed the start-up of the **UMP** store daemon (`umestoreds`) when run as a Microsoft Windows service. At startup, `umestoreds` creates a separate thread to communicate with the Microsoft Windows Service Manager while the cache file loads.
- Added the ability in **Ultra Messaging Manager** to configure the LBM license in the `umm.properties` file.
- Added **UMP** liveness detection creation and deletion callbacks to the sample applications, `umesrc.c`, `umesrc.cs` and `umesrc.java`.
- `umestored` now logs a warning if it releases a message and that message has not been consumed by at least one receiver whose state is maintained in `umestored`. The warning has the identifier: Store-6007-2.

### 8.2.3. Bug Fixes

- Resolved the Known Issue: *If a source application stops and you restart it before the receiving application declares the EOS event, the receiving application does not send a new keepalive message. The source requires a keepalive message in order to declare a receiver "alive."*
- Corrected a problem that caused a restarted store to delete source and topic information after a short period of time. A restarted store now correctly persists the source lifetime timeouts which prevents the source from deleting the source and topic.
- Fixed a memory leak that occurred in the **UMP** store daemon (`umestored`) when deleting proxy sources.
- Fixed a small memory leak on **UMP** sources related to named store resolution.
- Fixed a problem that could cause TSNI's with incorrect sequence numbers to be sent when a **UMP** source re-registers. This could trigger unrecoverable loss at non-**UMP** receivers.
- Changed a non-fatal assert [`node!=NULL`] to a debug statement stating the last sequence number in the ASL, and the sequence number currently searched on. For example: `[27461:1100937536|1041.662]:node is NULL. Last SQN in ASL 37e SQN number requested 37f`
- Corrected a problem with `umestored` that causes a receiver's state to be improperly created during store recovery.
- Corrected a problem that caused a store upon startup to delay persisting messages for up to the configured `retransmit_request_generation_interval` (`../Config/latejoinoptions.html#RECEIVERRETRANSMITREQUESTGENERATIONINTERVAL`) (default 10 seconds), while attempting to recover messages that the source has already released. Now source reports the unavailability of the requested messages to the store and the store resume persistence without delay.

### 8.2.4. Known Issues

- **UMP** stores sometimes fail to send message stability acknowledgements to sources for messages sent while the store was being reported unresponsive by the source. Informatica is investigating this problem.
- A **UMP** store daemon, `umestored`, may crash if you enable proxy sources and the daemon is configured with a very restrictive port range, (`transport_lbtrm_multicast_address_low` (`../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTLOW`) - `transport_lbtrm_multicast_address_high` (`../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTHIGH`),

`transport_tcp_port_low`

(`../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPLOW`) -

`transport_tcp_port_high`

(`../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPHIGH`), etc.). Informatica recommends that you use a wide port range (at least 1000 ports) in your **UM** configuration file if you enable proxy sources for `umestored`. Informatica is investigating this problem.

- 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. As a workaround, Informatica recommends that you enable `receiver_delivery_control_loss_check_interval` (`../Config/deliverycontroloptions.html#RECEIVERDELIVERYCONTROLLOSSCHECKINTERVAL`) 2500 in the **UM** configuration file, not the `umestored` XML configuration file. The value of 2500 assumes the default `*_nak_generation_interval`. See [Preventing Undetected Loss](#) (`../Config/grouprelated.html#PREVENTUNDETECTEDLOSS2`) for more information. Informatica is investigating this problem.
- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. Informatica is working on a solution to this problem.
- 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]`. Changing `fd_management_type` (`../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 8.3. UMQ 5.3

### 8.3.1. Updates

- **JMS Message Selectors** - Support for message selectors was added to Ultra Messaging JMS. The feature can also be used in other APIs by setting the `message_selector` (`../Config/majoroptions.html#RECEIVERMESSAGESELECTOR`) receiver option to a valid message selector string. This allows receivers to filter messages based on the message properties of each incoming message. See [Message Selectors](#) (`../jms_concepts/messageselectors.html`).
- **Automatic Reassignment on Receiver Failure** - A receiver that fails can now return with its original assignment ID and continue to receive queued messages in the correct order. To enable this feature, use the new **UM** configuration option, `umq_session_id` (`../Config/ultramessagingqueuingoptions.html#CONTEXTUMQSESSIONID`). See also, [Queue Session IDs](#) (`../UME/designing-queuing-applications.html#QUEUE-SES-IDS`).

- **Ultra Messaging JMS now supports UMP Session IDs.** - Ultra Messaging JMS now supports the use of **UMP** Session IDs. Implementation of this Ultra Messaging JMS feature involves the new `ConnectionFactory` configuration option, `use_ump_session_ids` (`../Config/ultramessagingjmsoptions.html#CONNECTIONFACTORYUSEUMPSESSIONIDS`). See also **UMP Session IDs** (`../jms_concepts/sessionids.html#SESSIONIDSUMP`).
- Updated the `umqsltool` (`../UME/umqsltool.html`) **UMQ SINC** log utility to understand two new **SINC** log event types new to this release (`RCV_REG` and `CTX_REG SINC`), and added printing a message's topic to the default dump tool.
- Added **C API** (`../API/index.html`) and **Java API** (`../JavaAPI/html/index.html`) supports for message selectors to the queue browser commands `lbm_rcv_umq_queue_msg_list()` and `LBMReceive.queueMessageList()`. The message selector can be used by setting the `message_selector` structure in the API function input parameter list. This allows the queue to control which messages get listed by using the message selector string to filter based on the message properties of each currently enqueued message.

### 8.3.2. Bug Fixes

- Changed the `createConnection()` methods in the `ConnectionFactory` to be thread safe. This problem caused various exceptions in multithreaded applications.
- Corrected a queue master election problem that prevented a former slave queue instance running on the Solaris SPARC platform (Solaris x86 was not affected) from correctly electing itself master in the event of a master queue instance failure.
- Corrected a problem with **Dead Letter Queue** that caused expired messages to be correctly placed on the **Master Queue's Dead Letter Queue** but not created on one or more **Slave Queue** instances.
- Corrected the `LBMMessage.queueIndexInfo()` method to now simply return null if no **UMQ** indexed queuing index information is present for a message, rather than throwing an exception. This matches the existing behavior of the **.NET UMQ API**.
- Changed the **Ultra Messaging JMS** example application `pong.sh` to ensure that the `JAVA_HOME` environment variable has been set.
- Corrected problems with the `QueueReceiver.java` example application. Not the application is consistent with the **JMS** interfaces and can be be recompiled with just the `jms.jar`.
- Fixed a problem that could cause a crash in **UM** applications with multi-threaded transports enabled if a multi-threaded transport context happened to re-use the request port from an earlier instance of an application and received a **UMQ** keepalive packet intended for the previous application.
- Fixed a possible deadlock in **Ultra Messaging JMS** that could occur when closing a session that had asynchronous consumers.
- Corrected a problem with the `umqsltool` (`../UME/umqsltool.html`) utility that could cause a crash when working with a **SINC** log file that contained an application set for which no receivers had ever registered.
- Corrected a problem in **Ultra Messaging JMS** that resulted in the truncation of **Unicode Text** messages with non-ASCII characters.
- Corrected a problem with multiple **JMS** producers sending on the same topic within the same `LBMContext` (`JMSConnection`). A single **UM** source now caches all **JMS** producers sending on the topic.



- Corrected a problem with JMS messages received in a durable subscriber sent by a native **UM** source which resulted in an exception. Now **UM** checks the source of a JMS message before determining and sending the proper response.
- Corrected a problem with the **UMQ** queue daemon (`umestored`) that caused a crash in some circumstances when it received a message resubmission for a message it already had.
- Corrected a problem with proxy source election that resulted in proxy sources to bounce between configured stores due to a short proxy election interval. This interval has been lengthened.
- Fixed a problem that could cause a fatal assertion when an queue browser receiver or observer received a message from the queue in response to a previously canceled message retrieval request.
- Fixed a problem that could cause a `[topic->rcv!=NULL]` fatal assert when deleting a receiver with an outstanding **UMQ** deregister event.
- Corrected a problem with the **UM** JMS method `setJMSType()` that caused `getJMSType()` to always return `NULL`.
- Fixed an issue that could sometimes cause a **UMQ** queue daemon (`umestored`) configured as a disk queue to improperly start up if it had previously received messages that contained application headers (e.g. messages with the total message lifetime option set on the source, messages with an index set, or most JMS messages).
- Fixed a **UMQ** issue that could cause a receiver's portion size to be ignored when assigning messages with indices to that receiver.
- Fixed a **UMQ** issue that could result in several non-fatal asserts, such as `Core-5688-8: WARNING: failed assertion [stream->reassembler.full_msg->len==total_len]` and `Core-5688-8: WARNING: failed assertion [stream->reassembler.first_sqn==first_sqn]`. A system crash was also possible.
- Fixed a **UMQ** issue that would prevent context registrations in the queue from succeeding when a new context registered on an IP and port that had already been in use by a previous context.

### 8.3.3. Known Issues

- The method, `getJMSRedelivered` for a message always returns `false` for redelivered messages when using Queue Session IDs along with the topic's `message-reassignment-timeout` set to 0 (zero).
- The **UM** configuration file specified for a queue cannot also contain source attributes for a store such as, `source_ume_store 127.0.0.1:14567` or `source_ume_store_name NYstore2`.
- When servicing large message retrieval requests from a queue browser, the queue may become busy enough that the receiver perceives it as unresponsive and re-registers with the queue, canceling the message retrieval request. Informatica recommends limiting large message retrievals to 1000 messages at a time, waiting for each retrieval to finish before beginning the next retrieval request

## 9. Release UMS 5.2.2 / UMP 5.2.2 / UMQ 5.2.2 - March -

# 2012

## 9.1. UMS 5.2.2

### 9.1.1. Updates

- Updated the UM Gateway to append its log file by default instead of starting with a new log file everytime. You now have the option to remove any previous logging information from the UM Gateway log file at your discretion.
- Implemented the following new Hot Failover features in the C API (`./API/index.html`), the Java API (`./JavaAPI/html/index.html`) and the .NET API (`./DotNetAPI/doc/Index.html`).
  - Hot Failover for Wildcard Receivers
  - 64-bit Hot Failover sequence numbers
  - Reduced size of Hot Failover message headers
  - Added `lbm_hf_src_send_rcv_reset()` to allow hot-failover receivers to reset their state.

All example applications have been updated with these new features.

Compatibility Issues:

- All normal Hot Failover (no gaps or optional messages) work fine across all versions.
- New Receivers can handle intentional gap and optional messages from any version.
- Pre-UM Core 5.1.1 receivers will not be able to handle intentional gap and optional messages from 5.1.1 sources.

### 9.1.2. Bug Fixes

- Corrected a file descriptor problem that caused a fatal assert if a file descriptor was closed before it was cancelled within **UM**.
- Fixed an issue with ordering Hot Failover intentional gaps that could, in rare cases involving loss, cause a segfault.
- Resolved a Topic Resolution issue where certain inputs on the topic resolution socket could cause a buffer overrun, sometimes resulting in a crash.
- Corrected the handling of socket errors, such as when a TCP socket could not be connected on a Unix system. This error results in the repetition of the following error message: `lbm_socket_recv recv/recvfrom: (107) Transport endpoint is not connected.`
- Fixed an issue where bad input on a TCP stream could cause a buffer overrun, sometimes resulting in a crash.
- Fixed an issue where a malformed context advertisement could cause a SEGV in certain cases.
- Fixed several places where a malformed packet could cause parsing to go into an infinite loop or cause a fatal assert.

- Changed the behavior of **UM** when it cannot locate the **UM** library. Previously, **UM** printed a warning. Now **UM** exits when it cannot locate the **UM** library.
- Corrected `lbm_msg_properties_get()` to set the length parameter to the actual length of the property, and to accept a length which is larger than the required length of the property.
- Corrected a problem with the UM Gateway that prevented a **UMP** store name from being properly resolved through two consecutive gateway peer hops.
- Removed a number of fatal asserts in the message processing path that could cause a process to abort due to bad network input.
- Fixed an issue when receiving fragmented messages containing message properties that could result in a fatal assert or incorrect message length. Although no fatal assert now occurs in this situation, **UM** ignores message properties if `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) is set to arrival order without reassembly (option value = 0).
- Corrected a problem that caused boolean message property values to be inverted when retrieved with the .NET API.
- Changed the parsing of the context, source and receiver scoped `ume_session_id` (`../Config/ultramessagingpersistenceoptions.html#CONTEXTUMESESSIONID`) setting to allow octal, decimal, or hexadecimal values (e.g. 01, 1, 0x1, respectively). Parsing previously assumed that all strings were in hexadecimal.

### 9.1.3. Known Issues

- When using the LBT-IPC Resource Manager (`../Design/lbm-objects.html#LBT-IPC-RESOURCE-MANAGER`) tool to view the currently allocated IPC transport sessions, the Session ID is presented in the reverse byte order from what is given to the application via the source string.
- Hot Failover is not currently supported with Multitransport Threads.
- Configuring sources with LBT-RU, source-side filtering, and adaptive batching may cause a crash. All three options must be set to cause the problem. Informatica is working to address this issue.
- When using the UM Gateway Informatica recommends that you **do not** enable the Multitransport Threads feature. Informatica does not support nor test the operation of Multitransport Threads across the UM Gateway.
- Multitransport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: LOG Level 5:  
LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen:`

`rdma_bind_addr` failed (`r=-1`). This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.

- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, UMS may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support gateway failover, MIM, persistence, queuing, JMS, or Ultra Load Balancing. Gateway support of these features is in development.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a `segfault`. Informatica is aware of this and has not observed any ill effects from this `segfault`.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

#### 9.1.4. Version Compatibilities

- Although UMS 5.0 applications should run well with LBM 4.2 UM Gateways, all Gateways should always run the same version.

## 9.2. UMP 5.2.2

### 9.2.1. New Features

- **Operations Guide** - Added the first edition of the **Ultra Messaging** Operations Guide. This first edition contains monitoring strategies and troubleshooting information for supporting the operation of **UMP**. Future editions will include more problem resolution scenarios and error message descriptions.

## 9.2.2. Updates

- Informatica has demonstrated operation with JDBC interfaces to MySQL™ and Oracle® databases. You may be able to use other JDBC databases, but Informatica has only tested with MySQL and Oracle.
- Changed how the **UMP** store behaves when it runs out of disk space. Previously, the store continually logged error messages in the `umestored` log. Now the store logs a warning (`[WARNING]: Store-5688-5265: [WARNING] aio_proactor GQCS: (112) There is not enough space on the disk`) and exits when it can no longer write to disk.

## 9.2.3. Bug Fixes

- Changed the severity level of the error message `disk-cache file ends prematurely` that appears in the **UMP** store log file when the store restarts. This is now only a warning.
- Fixed a compatibility problem between the **UMM** GUI and Java Version 1.7. The `AttributesDialog` was using the same method name as the `Dialog` object.
- Fixed an issue that could cause a **UMP** store daemon (`umestored`) to improperly delete receiver state information during shutdown.
- Fixed a problem that caused sources (persistent publishers) to ignore message stability acknowledgements from stores marked as inactive. Sources now process message stability acknowledgements from inactive stores.
- Corrected a problem with restarted stores that prevented the registration of proxy sources.
- Fixed an issue with wildcard receivers using event queues where under certain circumstances messages would not be automatically deleted. In this case, **UMP** does not send delivery confirmations to the source.

## 9.2.4. Known Issues

- **UMP** stores sometimes fail to send message stability acknowledgements to sources for messages sent while the store was being reported unresponsive by the source. Informatica is investigating this problem.
- A **UMP** store daemon, `umestored`, may crash if you enable proxy sources and the daemon is configured with a very restrictive port range, (`transport_lbtrm_multicast_address_low` (`../Config/transportlbtrm-networkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTLOW`) - `transport_lbtrm_multicast_address_high` (`../Config/transportlbtrm-networkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTHIGH`), `transport_tcp_port_low` (`../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPPORTLOW`) - `transport_tcp_port_high` (`../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPPORTHIGH`), etc.). Informatica recommends that you use a wide port range (at least 1000 ports) in your **UM** configuration file if you enable proxy sources for `umestored`. Informatica is investigating this problem.
- 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. As a workaround, Informatica recommends that you enable `receiver_delivery_control_loss_check_interval`

(../Config/deliverycontroloptions.html#RECEIVERDELIVERYCONTROLLOSSCHECKINTERVAL) 2500 in the **UM** configuration file, not the `umstored` XML configuration file. The value of 2500 assumes the default `*_nak_generation_interval`. See Preventing Undetected Loss (../Config/grouprelated.html#PREVENTUNDETECTEDLOSS2) for more information. Informatica is investigating this problem.

- If a source application stops and you restart it before the receiving application declares the EOS event, the receiving application does not send a new keepalive message. The source requires a keepalive message in order to declare a receiver "alive." Informatica is investigating this problem.
- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. Informatica is working on a solution to this problem.
- 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]`. Changing `fd_management_type` (../Config/majoroptions.html#CONTEXTFDMANAGEMENTTYPE) to `devpoll` prevents this problem. Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 9.3. UMQ 5.2.2

### 9.3.1. Updates

- **JMS Queue Browser** - Added support for this JMS specification for the C API (../API/index.html), the Java API (../JavaAPI/html/index.html) and the JMS API (../JMSAPI/html/index.html). The .NET API (../DotNetAPI/doc/Index.html) will be supported in a future release. The JMS Queue Browser allows the following. (See also Queue Browser (../UME/ume-ops-operational-view.html#QUEUEBROWSER).)
  - Retrieving a list of topics and application sets from a running queue daemon via the new `LBMContext.queueTopicList()` Java API call.
  - Retrieving a list of currently-enqueued message IDs from a running queue daemon via the new `LBMReceiver.queueMessageList()` Java API call.
  - Retrieving specific messages by message ID from a running queue daemon via the new `LBMReceiver.queueMessageRetrieve()` Java API call.
- Added the following **Ultra Messaging** configuration options to support the JMS Queue Browser.
  - `create_queue_browser_context` (../Config/ultramessagingjmsoptions.html#CONNECTIONFACTORYCREATEQUEUEBROWSERCONTEXT)

- `queue_browser_creation_delay` (`../Config/ultramessagingjmsoptions.html#CONNECTIONFACTORYQUEUEBROWSERCREATIONDELAY`)
  - `queue_browser_timeout` (`../Config/ultramessagingjmsoptions.html#CONNECTIONFACTORYQUEUEBROWSERTIMEOUT`)
  - `umq_require_queue_authentication` (`../Config/ultramessagingqueuingoptions.html#CONTEXTUMQREQUIREQUEUEAUTHENTICATION`)
- Added the `umestored daemon` option, `lbm-password-file` and the `queue` option, `require-client-authentication`, to support the requirement for clients to authenticate with the queue before performing Queue Browsing actions such as, listing all topics within a queue or retrieving a specific message by ID.
  - Added two new return values for `lbm_geterr()`.
    - `LBM_ENO_QUEUE_REG` returns when a message send by source, which is connected to both a store and a queue, fails because the source has lost registration with the queue.
    - `LBM_ENO_STORE_REG` returns when a message send by source, which is connected to both a store and a queue, fails because the source has lost registration with the store.

`LBM_EUMENOREG` has been changed to return when a message send by a source fails because the source has lost registration with either a single store, a single queue or both a store and a queue with which the source was simultaneously connected.

### 9.3.2. Bug Fixes

- Resolved the Known Issue: *When using multiple sending threads, a Queue (umestored) can deadlock. As a workaround, Informatica suggests that you specify only a single sending thread. Note that 1 sending thread is the current default. You can now use multiple sending threads within a queue.*
- Fixed a race condition that could cause a reclaim source event to be delivered before all stability source events were delivered for a particular UMQ message. This occurred if the message was sent to a queue configured with more than one queue group. (Sending to a single group containing multiple queue instances was not affected.)
- Corrected an issue in JMS when sending messages larger than 64K resulted in the error message, `Core-5688-4383: JNI detected an exception in (jni_src_cb:423): java.lang.NullPointerException.`

### 9.3.3. Known Issues

- The UM configuration file specified for a queue cannot also contain source attributes for a store such as, `source_ume_store 127.0.0.1:14567` or `source_ume_store_name NYstore2`.

## 10. Release UMS 5.2.1 / UMP 5.2.1 / UMQ 5.2.1 - February 2012

### 10.1. UMS 5.2.1

#### 10.1.1. Updated Features

- Changed Topic Resolution behavior to more fairly distribute topic advertisements (TIR) when `resolver_advertisement_send_immediate_response` (`../Config/resolveroperationoptions.html#SOURCERESOLVERADVERTISEMENTSSENDIMMEDIATERESPONSE`) has been disabled. Previously, not sending immediate responses to queries and wildcard queries resulted in some topics being un-advertised for long periods of time.
- Added an error log message to indicate when a response operation cannot be completed because `request_tcp_bind_request_port` (`../Config/requestnetworkoptions.html#CONTEXTREQUESTTCPBINDREQUESTPORT`) has been set to 0 (zero). This occurs for features that use the response channel, such as Late Join, Request/Response, Unicast Immediate Messages, **UMP** and **UMQ**.

#### 10.1.2. Bug Fixes

- Corrected a problem with the UM Gateway that resulted in wildcard receivers deafness when a wildcard receiver application stopped and restarted before the gateway's proxy wildcard receiver restarted. This could result in the loss of gateway proxy receivers for the individual topics matching the wildcard receiver pattern. The UM Gateway now ensures that an individual topic proxy receiver exists for a proxy wildcard receiver after it receives any topic advertisement (TIR) for the topic instead of just the first TIR.
- Corrected an incompatibility with **UMDS 1.1.1** introduced in **UMS/UMP/UMQ 5.2** that prevented **UMDS Receivers** from using Late Join. **UMDS 1.1.1** may now be used with **UMS/UMP/UMQ 5.2.1**.

#### 10.1.3. Known Issues

- When using the LBT-IPC Resource Manager (`../Design/lbm-objects.html#LBT-IPC-RESOURCE-MANAGER`) tool to view the currently allocated IPC transport sessions, the Session ID is presented in the reverse byte order from what is given to the application via the source string.
- Hot Failover is not currently supported with Multitransport Threads.
- Configuring sources with LBT-RU, source-side filtering, and adaptive batching may cause a crash. All three options must be set to cause the problem. Informatica is working to address this issue.
- When using the UM Gateway Informatica recommends that you **do not** enable the Multitransport Threads feature. Informatica does not support nor test the operation of Multitransport Threads across the UM Gateway.
- Multitransport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).



- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), UMS 4.1 issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, UMS may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support gateway failover, MIM, persistence, queuing, JMS, or Ultra Load Balancing. Gateway support of these features is in development.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

#### 10.1.4. Version Compatibilities

- Although UMS 5.0 applications should run well with LBM 4.2 UM Gateways, all Gateways should always run the same version.

## 10.2. UMP 5.2.1

### 10.2.1. Bug Fixes

No specific **UMP** problems were discovered.

### 10.2.2. Known Issues

- A **UMP** store daemon, `umestored`, may crash if you enable proxy sources and the daemon is configured with a very restrictive port range, (`transport_lbtrm_multicast_address_low` (`../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTLOW`) - `transport_lbtrm_multicast_address_high` (`../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTHIGH`), `transport_tcp_port_low` (`../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPPORTLOW`) - `transport_tcp_port_high` (`../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPPORTHIGH`), etc.). Informatica recommends that you use a wide port range (at least 1000 ports) in your **UM** configuration file if you enable proxy sources for `umestored`. Informatica is investigating this problem.
- 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. As a workaround, Informatica recommends that you enable `receiver_delivery_control_loss_check_interval` (`../Config/deliverycontroloptions.html#RECEIVERDELIVERYCONTROLLOSSCHECKINTERVAL`) 2500 in the **UM** configuration file, not the `umestored` XML configuration file. The value of 2500 assumes the default `*_nak_generation_interval`. See `Preventing Undetected Loss` (`../Config/grouprelated.html#PREVENTUNDETECTEDLOSS2`) for more information. Informatica is investigating this problem.
- If a source application stops and you restart it before the receiving application declares the EOS event, the receiving application does not send a new keepalive message. The source requires a keepalive message in order to declare a receiver "alive." Informatica is investigating this problem.
- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. Informatica is working on a solution to this problem.
- 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]`. Changing `fd_management_type` (`../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 10.3. UMQ 5.2.1

### 10.3.1. Bug Fixes

No specific UMQ problems were discovered.

### 10.3.2. Known Issues

- The UM configuration file specified for a queue cannot also contain source attributes for a store such as, `source_ume_store 127.0.0.1:14567` or `source_ume_store_name NYstore2`.
- When using multiple sending threads, a Queue (`umestored`) can deadlock. As a workaround, Informatica suggests that you specify only a single sending thread. Note that 1 sending thread is the current default.
- 

## 11. Release LBM 4.2.11 / UME 3.2.11 / UMQ 2.1.11 - February 2012

### 11.1. LBM 4.2.11

#### 11.1.1. Updated Features

- Changed Topic Resolution behavior to more fairly distribute topic advertisements (TIR) when `resolver_advertisement_send_immediate_response` (`../Config/resolveroperationoptions.html#SOURCERESOLVERADVERTISEMENTSSENDIMMEDIATERESPONSE`) has been disabled. Previously, not sending immediate responses to queries and wildcard queries resulted in some topics being un-advertised for long periods of time.
- Added an error log message to indicate when a response operation cannot be completed because `request_tcp_bind_request_port` (`../Config/requestnetworkoptions.html#CONTEXTREQUESTTCPBINDREQUESTPORT`) has been set to 0 (zero). This occurs for features that use the response channel, such as Late Join, Request/Response, Unicast Immediate Messages, **UMP** and **UMQ**.

#### 11.1.2. Bug Fixes

- Corrected a problem with the UM Gateway that resulted in wildcard receivers deafness when a wildcard receiver application stopped and restarted before the gateway's proxy wildcard receiver restarted. This could result in the loss of gateway proxy receivers for the individual topics matching the wildcard receiver pattern. The UM Gateway

now ensures that an individual topic proxy receiver exists for a proxy wildcard receiver after it receives any topic advertisement (TIR) for the topic instead of just the first TIR.

### 11.1.3. Known Issues

- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using `Automatic Monitoring` (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **UMS** may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- The UM Gateway does not currently support responses that exceed the maximum datagram size.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.

- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

## 11.2. UME 3.2.11

### 11.2.1. Bug Fixes

No specific UMP problems were discovered.

### 11.2.2. Known Issues

- Receivers using event queues and Spectrum with UMP can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]`. Changing `fd_management_type` (`./Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- UMP proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 11.3. UMQ 2.1.11

### 11.3.1. Bug Fixes

No specific UMQ problems were discovered.

## 12. Release UMS 5.2 / UMP 5.2 / UMQ 5.2 - December

# 2011

## 12.1. UMS 5.2

### 12.1.1. New Features

- **Off Transport Recovery (OTR)** - Provides recovery from brief unrecoverable loss at the transport level by accessing the source's retention buffer to re-request messages that no longer exist in a transport's transmission window. See [Off-Transport Recovery \(OTR\) \(../Design/lbm-features.html#OTR\)](#).

### 12.1.2. Updates

- **UM Gateway Improvements**
  - Added additional Web Monitor statistics and changed the display to indicate which statistics are subsumed into other statistics.
  - Implemented Use Query messages during gateway startup to determine receiver interest. Use Queries address both topics and wildcard patterns. Receivers respond with topic/pattern query responses and the gateway registers their interest. Previously, gateways relied on Topic Resolution Topic Queries and Wildcard Topic Queries which caused all UM applications to respond, spiking TR traffic. Mixing UM versions results in the old behavior. See [Interest and Use Queries \(../Gateway/concepts.html#INTEREST-AND-USE-QUERIES\)](#).
  - Implemented keepalives between gateway peer portals. You can configure the interval between keepalives, the inactivity time after which to consider the companion peer portal as unresponsive, and whether to send keepalives only when the portal is idle. See [Gateway Keepalive \(../Gateway/concepts.html#GATEWAY-KEEPALIVE\)](#).
- **Late Join Improvements** - Receivers now begin a Late join operation by sending a Late Join Initiation Request (LJIR). Sources respond with a Late Join Information (LJI) message containing updated information about what retained messages are available for retransmission. This method transmits better high and low sequence numbers to the receiver. For more see, [Using Late Join \(../Design/lbm-features.html#USING-LATE-JOIN\)](#).
- **ZOD source events** - Implemented Zero Object Delivery (ZOD) feature for Java and .NET receivers and sources. See [Zero Object Delivery \(ZOD\) \(../Design/design.html#ZOD-RECEIVER\)](#).
- Changed the name of the hash function, **default** to **classic** in both UMS and the UM Gateway. The value `default` is not supported and results in an error.
- Added `lbm_set_lbtrm_src_loss_rate()` and `lbm_set_lbtru_src_loss_rate()` in the C API ([../API/index.html](#)) to set an RM and RU source packet loss rate for testing purposes.
- Added the configuration option, `resolver_ud_acceleration` ([../Config/transportaccelerationoptions.html#CONTEXTRESOLVERUDACCELERATION](#)) to disable Accelerated Multicast for Topic Resolution. By default, Topic Resolution will no longer be accelerated unless you set this option to 1.
- Optimized the delivery of `LBMMessage` objects when using the .NET API.

- Increased the maximum size of all **UM** Web Monitor pages to 1 MB. The previous limit was 64K.
- Updated all receiving sample applications to print the same transport statistics previously provided by `lbmrcv`. This includes loss statistics in the bandwidth format `[loss, burst loss, unrecovered]` and retransmission statistics in the bandwidth format `-RX- num_rx_msgs`.
- Added several new `LBMSource.send()` methods to the Java API (`../JavaAPI/html/index.html`) that accept a direct `ByteBuffer`, a starting offset, and a length as parameters. Sends using a direct `ByteBuffer` can be slightly more efficient than sends from a `byte[]` array in some cases.

### 12.1.3. Bug Fixes

- Resolved the Known Issue: *When using the UM Gateway in a failover configuration using LBT-RDMA, if the gateways experience multiple failures, receivers may experience deafness (unable to discover sources)*. Now when using LBT-RDMA through the UM Gateway, a receiver can successfully failover from one path to another when there are multiple paths.
- Resolved the Known Issue: *The UM Gateway does not currently support responses that exceed the maximum datagram size*.
- Corrected a problem with the UM Gateway that resulted in an incorrect number of Use Queries. The number of Use Queries now matches the configured number.
- Fixed a problem bug that occurred if a new application connects via the same LBT-RU `ip:port` used by a recent application and `transport_source_side_filtering_behavior` (`../Config/majoroptions.html#SOURCETRANSPORTSOURCESIDEFILTERINGBEHAVIOR`) is enabled, the new application could receive messages in which it is uninterested.
- Corrected a problem that could cause an assert if an application joins an LBT-RU transport session that was previously joined by an application on the same `ip:port`.
- Corrected a Topic Resolution problem that occurred when using different `resolver_multicast_incoming_port` (`../Config/multicastresolvernetworkoptions.html#CONTEXTRESOLVERMULTICASTINCOMINGPORT`) and `resolver_multicast_outgoing_port` (`../Config/multicastresolvernetworkoptions.html#CONTEXTRESOLVERMULTICASTOUTGOINGPORT`) but the same `resolver_multicast_incoming_address` (`../Config/multicastresolvernetworkoptions.html#CONTEXTRESOLVERMULTICASTINCOMINGADDRESS`) and `resolver_multicast_outgoing_address` (`../Config/multicastresolvernetworkoptions.html#CONTEXTRESOLVERMULTICASTOUTGOINGADDRESS`). As a result, queries could be suppressed which leads to receiver deafness.
- Corrected a problem with the UM Gateway that resulted in gateways continuing to forward messages on topics for which remote interest no longer exists. Gateways now stop forwarding these messages after the configured timeout.
- Corrected a problem that removed commas from configuration option values if the options were set with an XML configuration file.
- Fixed a problem on AIX that prevented Event Queues from holding more than 32K of items and events.
- Fixed an issue with the Java API that caused application headers to be removed from an `LBMessage` when delivered to multiple callbacks on a machine with memory alignment.

- Corrected a problem with XML configuration files that resulted in a parse error when the element <options> did not contain a child element.
- Fixed a problem with Source Side Filtering that could result in a fatal assert when using Multithreaded Transports (MTT). The fatal assertion was removed since with MTT, a NULL resolver pointer is an acceptable condition.
- Corrected a problem with XML configurations that prevented the full range of values for certain configuration options from being specified under allow or deny rules, depending on the platform used. The full range of values can now be specified for all platforms.
- Corrected the **UM C** sample applications to be able to parse rate limits greater than 32 bits set on the command line.
- Fixed several problems that could cause an LBM application to crash if it received various types of malformed or unexpected packets.
- Experimental licenses for UD Acceleration and LBT-RDMA are no longer reported as invalid.
- Corrected a problem with the UM Gateway that could cause loss to be reported for messages sent by a source on one side of the gateway prior to the creation of a receiver on the other side.
- Corrected a problem with .NET **UM** initialization that could result in Microsoft Windows socket APIs being invoked prior to calling `WSAStartup`.
- Corrected a problem with the UM Gateway that resulted in a retransmission being sent more than once to a receiver when multiple paths exist from a source to the receiver and late join was enabled.
- Resolved an issue with the UM Gateway that resulted in a core dump in receiving applications when forwarding a message containing message properties.
- Corrected a problem that arose from creating Wildcard Receivers with the same pattern but different `pattern_types` (`./Config/wildcardreceiveroptions.html#WILDCARDRECEIVERPATTERNATYPE`). **UM** now supports identical patterns with different types in Topic Resolution and across the UM Gateway.
- Corrected a problem with the UM Gateway that occurred when using **UMP** across multiple gateway hops. Restarting one of the gateways resulted in receiver deafness.
- Corrected a problem that occurred when creating and canceling timers that use event queues. **UM** cancelled the wrong timer if it was already dispatched to the event queue.
- Corrected a problem that occurred when receiving a message containing message properties with ZOD enabled in Java could result in a `java.lang.IllegalArgumentException` or a `java.nio.BufferUnderflowException`.

#### 12.1.4. Known Issues

- When using the LBT-IPC Resource Manager (`./Design/lbm-objects.html#LBT-IPC-RESOURCE-MANAGER`) tool to view the currently allocated IPC transport sessions, the Session ID is presented in the reverse byte order from what is given to the application via the source string.
- Hot Failover is not currently supported with Multi-Transport Threads.
- Configuring sources with LBT-RU, source-side filtering, and adaptive batching may cause a crash. All three options must be set to cause the problem. Informatica is working to address this issue.



- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): `vms_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using `Automatic Monitoring` (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **UMS** may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support gateway failover, MIM, persistence, queuing, JMS, or Ultra Load Balancing. Gateway support of these features is in development.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

### 12.1.5. Version Compatibilities

- Although **UMS 5.0** applications should run well with **LBM 4.2** UM Gateways, all Gateways should always run the same version.

## 12.2. UMP 5.2

### 12.2.1. New Features

- **Liveness Detection** - Implemented the ability to track the "liveness" of a receiver application. See Receiver Liveness Detection ([../UME/designing-persistent-applications.html#RECEIVER-LIVENESS-DETECTION](http://../UME/designing-persistent-applications.html#RECEIVER-LIVENESS-DETECTION))

### 12.2.2. Updates

- **Ultra Messaging Manager (UMM) Improvements**
  - **One-time Table Update** - Existing **UMM** users must run a one-time script to update your database tables. Please run `UMM/update_mysql_tables.sql` to change the constraints on the application table. Also the file, `UMM/templates.txt` has been renamed to `UMM/mysql_templates.txt`.
  - Added JDBC Support for the **UMM** database. You can now use an Oracle database or a MySQL database. See Configuring the UMM Database ([../UMM/umm-daemon.html#UMM-DB](http://../UMM/umm-daemon.html#UMM-DB)).
  - Implemented SSL encryption between the UMM Daemon and UM applications. See Securing UMM Daemon Communication with SSL ([../UMM/umm-daemon.html#UMM-SSL](http://../UMM/umm-daemon.html#UMM-SSL)).
- **Clean Shutdown** - The **UMP** store and **UMQ** queue daemon (`umestored`) now support a **clean** shutdown mode. See `umestored` ([../UME/umestored-manpage.html](http://../UME/umestored-manpage.html)).

### 12.2.3. Bug Fixes

- Fixed a problem that allowed **UMP** proxy sources to steal Registration IDs from an active source. **UMP** stores now ignore registration requests from proxy sources while the store is still registered with an active source.
- Corrected a problem when setting the configuration option, `ume_session_id` ([../Config/ultramessagingpersistenceoptions.html#SOURCEUMESESSIONID](http://../Config/ultramessagingpersistenceoptions.html#SOURCEUMESESSIONID)) with a hexadecimal value. **UM** now correctly parses any valid hexadecimal value specified for this option.
- Removed the **Ctrl-z** keyboard accelerator on the exit menu of the **UMM** GUI.
- Fixed a Persistence problem caused by network loss of messages, followed by a timeout and subsequent re-registration of the source, which lead to messages being unrecoverably lost without notification to the receiver. Unrecoverable loss is now correctly reported.

### 12.2.4. Known Issues

- A **UMP** store daemon, `umestored`, may crash if you enable proxy sources and the daemon is configured with a very restrictive port range, (`transport_lbtrm_multicast_address_low` ([../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTLOW](http://../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTLOW)) - `transport_lbtrm_multicast_address_high`

- (../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMSOURCEPORTHIGH),  
 transport\_tcp\_port\_low  
 (../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPLOW) -  
 transport\_tcp\_port\_high  
 (../Config/transporttcpnetworkoptions.html#CONTEXTTRANSPORTTCPHIGH), etc.). Informatica recommends that you use a wide port range (at least 1000 ports) in your **UM** configuration file if you enable proxy sources for `umestored`. Informatica is investigating this problem.
- 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. As a workaround, Informatica recommends that you enable `receiver_delivery_control_loss_check_interval` (../Config/deliverycontroloptions.html#RECEIVERDELIVERYCONTROLLOSSCHECKINTERVAL) 2500 in the **UM** configuration file, not the `umestored` XML configuration file. The value of 2500 assumes the default `*_nak_generation_interval`. See Preventing Undetected Loss (../Config/grouprelated.html#PREVENTUNDETECTEDLOSS2) for more information. Informatica is investigating this problem.
  - If a source application stops and you restart it before the receiving application declares the EOS event, the receiving application does not send a new keepalive message. The source requires a keepalive message in order to declare a receiver "alive." Informatica is investigating this problem.
  - Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. Informatica is working on a solution to this problem.
  - 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]. Changing `fd_management_type` (../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE) to `devpoll` prevents this problem. Informatica is investigating this problem.
  - **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 12.3. UMQ 5.2

### 12.3.1. Updates

- **Ultra Messaging Manager (UMM) Improvements** - Added **UMM** GUI support for JMS applications. See Creating JMS Configurations (../UMM/umm-gui.html#UMMGUI-JMS).
- **Additional Ultra Messaging JMS XML Configuration Format** - Added the ability to configure Ultra Messaging JMS using the same XML format used by the **UMM** GUI. The original `<JMSConfig>` format is still valid. See Ultra Messaging JMS Configuration (../jms\_concepts/umjms-config.html).
- **Queue Status** - Implemented the **UMQ** SINC Log Tool for comparison and analysis of **UMQ** log files. See `umqsltool` (../UME/umqsltool.html).

- **Index Reservation API** - Added `lbm_rcv_umq_index_reserve()` in the C API (`./API/index.html`) for **UMQ** and ULB receivers to reserve specific indices for future assignment. This API can be used for indexed queuing and indexed ULB.
- Added the configuration options, `RECEIVER_CREATION_DELAY` (`./Config/ultramessagingjmsoptions.html#CONNECTIONFACTORYRECEIVERCREATIONDELAY`), `SOURCE_CREATION_DELAY` (`./Config/ultramessagingjmsoptions.html#CONNECTIONFACTORYSOURCECREATIONDELAY`), and `SOURCE_REGISTRATION_TIMEOUT` (`./Config/ultramessagingjmsoptions.html#CONNECTIONFACTORYSOURCEREGISTRATIONTIMEOUT`), to handle source and receiver creation options. These options should always be used in uppercase in a **UM** configuration file.
- Improved the performance of Ultra Messaging JMS internal logging code in source event callbacks.

### 12.3.2. Bug Fixes

- Fixed a problem that caused the **UMQ** queue daemon (`umestored`) to access free'd memory.
- Fixed a problem with the reception of large messages (usually 8k or larger) that included message properties by JMS and Java receivers. These receivers can now handle large messages with message properties.
- Corrected a problem that prevented Registration IDs from being set correctly for some JMS destinations.
- Removed the trimming of the message payload for JMS `TextMessages`. Now, JMS does not trim any characters from `TextMessage` strings.
- Fixed a problem that resulted in application headers used for Indexed Queuing and Message Lifetimes being created incorrectly. Previously, either Indexed Queuing and Message Lifetimes were usable separately but not simultaneously. Now both are available provided you use Version 5.2 of both the **UM** library and the `Queue` (`umestored`).
- Fixed an issue that could cause **UMQ** receivers to go deaf to a queue if the queue was killed and restarted rapidly after the receiver finished registration.
- Corrected a problem that may have caused transactions across Sessions to adversely affect each other. An internal list of sources is now in the Session object.
- Fixed an issue with Ultra Messaging JMS that prevented the JMS message id from being set correctly when ULB was specified in the source attributes.
- Corrected a problem with Ultra Messaging JMS that prevented a message acknowledgement if the session was set to `CLIENT_ACKNOWLEDGE` mode.
- Corrected a timing problem in Ultra Messaging JMS that could result in the receiver being closed before the last message was acknowledged.

### 12.3.3. Known Issues

- The **UM** configuration file specified for a queue cannot also contain source attributes for a store such as, `source_ume_store 127.0.0.1:14567` or `source_ume_store_name NYstore2`.

- When using multiple sending threads, a Queue (`umestored`) can deadlock. As a workaround, Informatica suggests that you specify only a single sending thread. Note that 1 sending thread is the current default.

## 13. Release UMS 5.1.1 / UMP 5.1.1 / UMQ 5.1.1 - November 2011

### 13.1. UMS 5.1.1

#### 13.1.1. Bug Fixes

No specific UMS problems were discovered.

#### 13.1.2. Known Issues

- With `ud_acceleration` (`./Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) turned ON for VMA support in Infiniband, receivers are unable to resolve sources from within the same process. Similarly, sources are unable to discover themselves within the same process. No resolution issues exist if sources and receivers run in different process. Informatica is investigating this issue.
- Late join initiated from receivers in a context configured for Multi-Transport Threads (MTT) will not work through the UM Gateway if the gateway is configured to forward late join traffic. Informatica is investigating this issue.
- Hot Failover is not currently supported with Multi-Transport Threads.
- Configuring sources with LBT-RU, source-side filtering, and adaptive batching may cause a crash. All three options must be set to cause the problem. Informatica is working to address this issue.
- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`./Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`./Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), UMS 4.1 issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen:`

`rdma_bind_addr` failed (`r=-1`). This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.

- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, UMS may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- The UM Gateway does not currently support responses that exceed the maximum datagram size.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

### 13.1.3. Version Compatibilities

- Although UMS 5.0 applications should run well with LBM 4.2 UM Gateways, all Gateways should always run the same version.

## 13.2. UMP 5.1.1

### 13.2.1. Bug Fixes

No specific UMP problems were discovered.

### 13.2.2. Known Issues

- Receivers using event queues and Spectrum with UME can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching

event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.

- 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]`. Changing `fd_management_type` (`../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 13.3. UMQ 5.1.1

### 13.3.1. Bug Fixes

- Fixed an issue that caused a JMS exception. As a result, a JMS message can now be acknowledged even after it has been modified.

## 14. Release LBM 4.2.10 / UME 3.2.10 / UMQ 2.1.10 - November 2011

### 14.1. LBM 4.2.10

#### 14.1.1. Bug Fixes

- Corrected a problem with the UM Gateway that resulted in wildcard receivers deafness to some but not all topics that fit the wildcard receiver's topic pattern. This only occurred when the wildcard receivers were accessible via a peer link, and wildcard receivers with the identical pattern existed in at least 2 other domains.

#### 14.1.2. Known Issues

- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`./Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`./Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), UMS 4.1 issues the following warning: LOG Level 5:  
LBT-RDMA: VMS Logger Message (Error): vmss\_create\_store: 196[E] vms\_listen:  
rdma\_bind\_addr failed (r=-1). This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`./Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`./Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, UMS may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- The UM Gateway does not currently support responses that exceed the maximum datagram size.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

## 14.2. UME 3.2.10

### 14.2.1. Bug Fixes

No specific UMP problems were discovered.



## 14.2.2. Known Issues

- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use LBM\_EVQ\_BLOCK when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]. Changing `fd_management_type` (`./Config/majoroptions.html#CONTEXTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 14.3. UMQ 2.1.10

### 14.3.1. Bug Fixes

No specific UMQ problems were discovered.

## 15. Release UMS 5.1 / UMP 5.1 / UMQ 5.1 - November 2011

### 15.1. UMS 5.1

#### 15.1.1. New Features

- Added the message property object that allows your application to insert named, typed metadata to topic messages and implement functionality that depends on the message properties. **UMS** allows eight property types: boolean, byte, short, int, long, float, double, and string. See [Message Properties Object](#) (`./Design/lbm-objects.html#MESSAGEPROPERTIESOBJECT`) for more.

#### 15.1.2. Updated Features

- Added the functions, `lbm_hf_rcv_topic_dump()`, `lbm_hfx_attr_dump()`, `lbm_hfx_dump()`, and `lbm_hfx_rcv_topic_dump()`, in the C API (`./API/index.html`) to dump the attributes of HFXs, HFX receivers,

and hot failover receivers. Also added corresponding functions in the Java API (`../JavaAPI/html/index.html`) and .NET API (`../DotNetAPI/doc/Index.html`) to dump attributes of HFXs and HFX receivers.

### 15.1.3. Bug Fixes

- Improved multi-thread locking algorithms for Multi-Transport Threads to eliminate the possibility of receiver (or wildcard receiver) creation or deletion returning error EOP due to locking contention of the transport threads. Lock contention no longer requires an application to retry receiver creation or deletion.
- Corrected a memory leak in the UM Gateway that occurred when an endpoint portal was unable to send a message.
- Corrected a problem with the UM Gateway that occurred if the gateway was stopped and restarted. In this event, certain unicast control messages to destination contexts which were currently unknown to the gateway were dropped, but no effort was made to locate those destination contexts. Now those messages are still dropped, but the gateway sends queries to locate the destination context.

### 15.1.4. Known Issues

- With `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) turned ON for VMA support in Infiniband, receivers are unable to resolve sources from within the same process. Similarly, sources are unable to discover themselves within the same process. No resolution issues exist if sources and receivers run in different process. Informatica is investigating this issue.
- Late join initiated from receivers in a context configured for Multi-Transport Threads (MTT) will not work through the UM Gateway if the gateway is configured to forward late join traffic. Informatica is investigating this issue.
- Hot Failover is not currently supported with Multi-Transport Threads.
- Configuring sources with LBT-RU, source-side filtering, and adaptive batching may cause a crash. All three options must be set to cause the problem. Informatica is working to address this issue.
- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): vmss\_create\_store: 196[E] vms\_listen: rdma\_bind\_addr failed (r=-1). This warning indicates that `rdma_bind` failed for ethernet interfaces,

which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.

- When using `Automatic Monitoring` (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, UMS may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- The UM Gateway does not currently support responses that exceed the maximum datagram size.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

### 15.1.5. Version Compatibilities

- Although UMS 5.0 applications should run well with LBM 4.2 UM Gateways, all Gateways should always run the same version.

## 15.2. UMP 5.1

### 15.2.1. New Features

- Enabled the use of the UMM API that allows you to programmatically create and store application configurations in the UMM database. The UMM GUI uses the same API to create users, passwords, applications configurations and configuration templates. See UMM Java API (`../UMMJavaAPI/html/index.html`) for UMM objects, constructors and methods.

### 15.2.2. Bug Fixes

- Corrected a problem with the **UMM** GUI Permissions dialog box that caused the dialog box to be deleted after rearranging the column headings.

### 15.2.3. Known Issues

- Receivers using event queues and Spectrum with **UME** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]`. Changing `fd_management_type` (`./Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 15.3. UMQ 5.1

### 15.3.1. Updated Features

- Added support for the JMS message properties specification through the **UM** JMS API (`./JMSAPI/html/index.html`).

### 15.3.2. Bug Fixes

- Fixed an issue with **UMQ** indexed queuing from a Java source that prevented the queue from sending correctly which produced the following exception: `com.latencybusters.lbm.LBMEInvalException: UMQ index too long.`

## 16. Release LBM 4.2.9 / UME 3.2.9 / UMQ 2.1.9 -

# November 2011

## 16.1. LBM 4.2.9

### 16.1.1. Bug Fixes

- Corrected a problem with the UM Gateway that resulted in wildcard receivers deafness when a source matching the wildcard pattern was stopped and then restarted.
- Corrected a problem with the UM Gateway that resulted in wildcard receivers deafness to some of the topics it previously matched and received.

### 16.1.2. Known Issues

- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`./Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`./Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: LOG Level 5:  
LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`./Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`./Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **UMS** may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- The UM Gateway does not currently support responses that exceed the maximum datagram size.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.

- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

## 16.2. UME 3.2.9

### 16.2.1. Bug Fixes

No specific **UMP** problems were discovered.

### 16.2.2. Known Issues

- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]. Changing fd_management_type (./Config/majoroptions.html#CONTEXTFDMANAGEMENTTYPE) to devpoll prevents this problem. Informatica is investigating this problem.`
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 16.3. UMQ 2.1.9

### 16.3.1. Bug Fixes

No specific **UMQ** problems were discovered.

## 17. Release UMS 5.0.1 / UMP 5.0.1 / UMQ 5.0.1 - September 2011

### 17.1. UMS 5.0.1

#### 17.1.1. Bug Fixes

- Renamed the hash function, `default` to `classic`. As a result, `default` is no longer a valid selection for **UMS** or **UM Gateway** hash functions. However if you do specify `default`, an error message appears and **UMS** or **UM Gateway** uses the new default hash function, `murmur2`.
- Corrected a problem that may cause an application to crash due to the use of initialized data. This problem occurred when a newly created Hot Failover receiver and an existing source joined an active transport. **UMS** now initializes data prior to the creation of underlying receiver objects which prevents this condition from occurring.
- Enhanced the **UM Gateway** to speed topic resolution by propagating interest immediately upon detection of an adjacent, active gateway.
- Corrected an error in handling **UM Gateway ACL** port comparisons.
- Corrected an error in the **UM Gateway** that caused receiver deafness when using **UMP** across multiple gateway hops and one of the gateways was restarted.
- Corrected a receiver deafness problem that arose when using `transport_source_side_filtering_behavior` (`../Config/majoroptions.html#SOURCETRANSPORTSOURCESIDEFILTERINGBEHAVIOR`) if the `request_tcp_interface` (`../Config/requestnetworkoptions.html#CONTEXTREQUESTTCPINTERFACE`) and the `transport_lbtru_interface` (`../Config/transportlbt-runetworkoptions.html#SOURCETRANSPORTLBTRUINTERFACE`) were different.
- Corrected a problem that caused a crash if you enable both `transport_source_side_filtering_behavior` (`../Config/majoroptions.html#SOURCETRANSPORTSOURCESIDEFILTERINGBEHAVIOR`) and **Multi-Transport Threads** (`use_transport_thread` (`../Config/majoroptions.html#RECEIVERUSETRANSPORTTHREAD`)) and a receiver sends the `SSF include` list to the source.

#### 17.1.2. Known Issues

- Late join initiated from receivers in a context configured for **Multi-Transport Threads (MTT)** will not work through the **UM Gateway** if the gateway is configured to forward late join traffic. Informatica is investigating this issue.
- **Hot Failover** is not currently supported with **Multi-Transport Threads**.
- Configuring sources with **LBT-RU**, **source-side filtering**, and **adaptive batching** may cause a crash. All three options must be set to cause the problem. Informatica is working to address this issue.
- When using the **UM Gateway** Informatica recommends that you **do not** enable the **Multi-Transport Threads** feature. Informatica does not support nor test the operation of **Multi-Transport Threads** across the **UM Gateway**.

- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`./Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`./Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`./Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`./Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **UMS** may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- The UM Gateway does not currently support responses that exceed the maximum datagram size.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

### 17.1.3. Version Compatibilities

- Although **UMS 5.0** applications should run well with **LBM 4.2** UM Gateways, all Gateways should always run the same version.



## 17.2. UMP 5.0.1

### 17.2.1. Bug Fixes

No specific UMP problems were discovered.

### 17.2.2. Known Issues

- Receivers using event queues and Spectrum with UME can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use LBM\_EVQ\_BLOCK when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]. Changing fd_management_type (./Config/majoroptions.html#CONTEXTFDMANAGEMENTTYPE) to devpoll prevents this problem.` Informatica is investigating this problem.
- UMP proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 17.3. UMQ 5.0.1

### 17.3.1. Bug Fixes

No specific UMQ problems were discovered.

## 18. Release UMS 5.0 / UMP 5.0 / UMQ 5.0 - August 2011

**Attention - Name Change:** Beginning with this release, the LBM product has been renamed to **Ultra Messaging Streaming Edition (UMS)** and the UME product has been renamed to **Ultra Messaging Persistence Edition (UMP)**. UMQ retains its name, **Ultra Messaging Queuing Edition (UMQ)**. Also with this release, the version number for these core UM products will be the same.

## 18.1. UMS 5.0

### 18.1.1. New Features

- **Documentation TOC and Search Page** - The UM Readme file is now presented in a framed, HTML page. The left pane contains a Table of Contents for all documentation and a search function box, providing easier access to the UM documentation.
- **XML Configuration Files** - Added the ability to use XML configuration files. See XML Configuration Files ([../Config/xmlconfigurationfiles.html](#)).
- **Pre-Defined Messaging (PDM)** - Added Pre-Defined Messaging (PDM) that provides an API similar to the SDM API that allows you to define messages once and then use the definition to create messages that may contain self-describing data. See Pre-Defined Messaging ([../Design/lbm-features.html#PRE-DEFINED-MESSAGING](#)).

### 18.1.2. Updates

- **Multiple Unicast Resolver Daemons (lbmrd)** - Added the ability to specify multiple Unicast Resolver Daemons (`lbmrd`), to be used in a warm-warm failover mode. When using multiple `lbmrds`, all applications should specify the same set of `lbmrds`. See Unicast Topic Resolution Resilience ([../Design/architecture.html#UNICAST-TOPIC-RES-RESILIENCE](#)).
- **Optimize Implicit Batching** - Changed the implicit batching feature to use a fixed-size array instead of a queue that was not allowed to exceed a predefined maximum size. Preallocating an array of the maximum size removes a malloc/free combination from the message path for every message. This change improves performance by about 30 percent.
- Added the `LBMObjectRecycler` and `LBMObjectRecyclerBase` classes in the Java API ([../JavaAPI/html/index.html](#)) and the .NET API ([../DotNetAPI/doc/Index.html](#)) to provide a way to reuse objects and reduce garbage collection. These classes support all of the Statistics types as well as the `LBMMessage` type so promoted messages can be recycled. The `lbmhfxrcv`, `lbmsrc`, `lbmpong`, `lbmrcv`, `lbmreq`, `lbmresp`, `lbmsrc`, `umercv`, `umesrc`, `umqrcv`, and `umqsrc` Java example applications (`../java_example`) and .NET example applications (`../dotnet_example`) have been updated to demonstrate the use of these classes.
- Added the **murmur2** hash function as a built-in option to the `resolver_string_hash_function` ([../Config/resolveroperationoptions.html#CONTEXTRESOLVERSTRINGHASHFUNCTION](#)) context configuration option. **murmur2** can perform better than the existing built-in hash functions for long topic strings.
- Added support for setting **UMS** wildcard receiver options to the UM Gateway.
- Added the public `LBMMessage.promote()` method. If applications intend to hold messages for later processing, they should call `promote()` to guarantee that messages are not reclaimed by **UMS**.
- Added `getAttributeValue` and `setAttributeValue` methods to `LBMHFXReceiver` to provide the same functionality as other **UM** objects.
- Added a new UM Gateway Web Monitor statistic to show the number of retransmit requests received by an endpoint for an unknown source.

### 18.1.3. Bug Fixes

- The UM Gateway now logs an error message if the initialization of gateway monitoring fails.
- Corrected a problem with Automatic Monitoring that caused incorrect clean up of implicitly spawned reactor contexts. Other problems fixed occurred when using `epoll` with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) causing the UD Acceleration library to report errors during clean up. When UD Acceleration was not used, the base `epoll` file descriptor leaked.
- In the UM Gateway, corrected a memory leak of 64 bytes for each `<regex-pattern>` in an ACL.
- Fixed a condition where a regular receiver's callback function could be called before the user application's `lbm_rcv_t` pointer is set.
- Added the `-E` option to the `lbmmrcv.java` example application to allow it to exit upon receiving an EOS like the C example application.
- Corrected a problem in the UM Gateway that occurred if you create a context with `request_tcp_bind_request_port` (`../Config/requestnetworkoptions.html#CONTEXTREQUESTTCPBINDREQUESTPORT`) = 0, and a source created on that context provides `late_join` (`../Config/latejoinoptions.html#SOURCELATEJOIN`). This configuration disabled Late Join. Now `lbm_src_topic_alloc()` returns the following error: `late join not allowed when request_tcp_bind_request_port is disabled`.
- Corrected a problem in the UM Gateway that caused errors relating to the creation of unicast forwarding associations to be incorrectly reported.
- Fixed an issue that could lead to a crash if implicit batching was being used and a particular OS socket send call failed, which is relatively rare failure.
- Fixed an issue that could cause a crash if using Adaptive Batching and you delete a source with messages still outstanding in the batch buffer.
- Fixed an issue in the Java API (`../JavaAPI/html/index.html`) that caused a `NullPointerException` if you passed in null for the third parameter to the `LBMHFXReceiver` constructor.
- Corrected a problem in the UM Gateway that caused deafness in wildcard receivers due to case-insensitive topic patterns. The UM Gateway now uses case-sensitive string compares when mapping topics and patterns.
- Corrected a problem in the UM Gateway that caused receivers using TCP, IPC, or LBT-RDMA transports to encounter the warning assertion: `MUL_SQN_GT(sq,ctrl->stream_high_sqn)` and receive data out of order despite Ordered Delivery was enabled.
- Removed a lock taken during a call to `lbm_context_delete()` which could cause a running IPC or RDMA receiver thread to deadlock during context deletion.
- Fixed an issue with the Java API (`../JavaAPI/html/index.html`) where setting both a context source event callback via `LBMContextAttributes.setContextSourceEventCallback` and also enabling receipt of topicless immediate messages via calling `LBMContext.enableImmediateMessageReceiver` could cause a crash upon receipt of a topicless immediate message. Setting only one (or none) of the two callbacks was successful, and setting both callbacks was successful if the newer, preferred `LBMContextAttributes.setImmediateMessageCallback` method was used to set the topicless immediate message callback instead of `LBMContext.enableImmediateMessageReceiver`.
- Optimized memory usage of the resolver cache for contexts on a network with **LBM 3.6** or earlier sources.

- Corrected a problem that caused the C APIs (`./API/index.html`) `lbm_*_attr_dump()` and their Java API (`./JavaAPI/html/index.html`) / .NET API (`./DotNetAPI/doc/Index.html`) equivalents to return strings containing garbage values for deprecated or unsupported options.
- Updated the version of `libpre` bundled with the AIX install packages to match the version of `libpre` Informatica uses to build the AIX **UM** packages.

#### 18.1.4. Known Issues

- Hot Failover is not currently supported with Multi-Transport Threads.
- Configuring sources with LBT-RU, source-side filtering, and adaptive batching may cause a crash. All three options must be set to cause the problem. Informatica is working to address this issue.
- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`./Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`./Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`./Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`./Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **UMS** may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- The UM Gateway does not currently support responses that exceed the maximum datagram size.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.

- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (transport\*\_activity\_timeout defaults to 60 seconds for LBTRU and LBTRM).

## 18.2. UMP 5.0

### 18.2.1. New Features

- **Ultra Messaging Manager** - This new feature ensures the consistency and reliability of enterprise production applications by enabling IT administrators to control what configurations messaging applications can use and what users can operate them. See Ultra Messaging Manager (../UMM/index.html).

### 18.2.2. Updates

- **Batching Acknowledgments** - Added the ability configure **UMP** to acknowledge message consumption to a store(s) for a series of messages independent of when the receiving application consumed the messages. This option works well if multiple threads process messages off of an event queue, which may result in messages being consumed out of order. See [Batching Acknowledgments](#) (../UME/designing-persistent-applications.html#BATCHINGACKS).
- **Object-free Explicit Acknowledgments** - When using explicit ACKs in a Java (../JavaAPI/html/index.html) or .NET (../DotNetAPI/doc/Index.html) application, you can enable **Zero Object Delivery (ZOD)** (<https://communities.informatica.com/infakb/faq/5/Pages/80233.aspx>) and extract ACK information from messages and then send acknowledgements to the store(s) for any sequence number. See [Object-free Explicit Acknowledgments](#) (../UME/designing-persistent-applications.html#OBJECT-FREEEXPLICITACKS).
- Added new configuration option, `ume_allow_confirmed_delivery` (../Config/ultramessagingpersistenceoptions.html#RECEIVERUMEALLOWCONFIRMEDDELIVERY).

### 18.2.3. Bug Fixes

- `lbn_msg_ume_send_explicit_ack()` now returns 0 (zero) for success instead of the number of bytes sent. It still returns -1 for failure.
- Fixed a problem originating from a source configured to use both **UMP** and ULB at the same time. This configuration caused the following assert: `WARNING: failed assertion [buff->umesendxcd!=NULL] at line 2543 in .../.../.../src/lib/lbn/lbnsrc.c.`

- Corrected a problem that appeared if you did not configure any **UMP** stores and set either `ume_confirmed_delivery_notification` (`../Config/ultramessagingpersistenceoptions.html#SOURCEUMECONFIRMEDDELIVERYNOTIFICATION`) or `ume_message_stability_notification` (`../Config/ultramessagingpersistenceoptions.html#SOURCEUMEMESSAGESTABILITYNOTIFICATION`) to deliver per message notifications as opposed to merely per fragment notifications.
- Fixed a rare condition that could briefly cause a single **UMP** receiver to be registered with a store multiple times for the same source.
- Corrected a fatal assertion that occurred when sending a combination of fragmented and non fragmented hot failover messages when running a Java receiver with `ume_use_ack_batching` (`../Config/ultramessagingpersistenceoptions.html#RECEIVERUMEUSEACKBATCHING`) enabled. The fatal assertion was `[MUL_SQN_GTE(first,ctrl->expected_sqn)]` at line 1555 in `../../../../src/lib/lbm/ume/umerctrl.c`
- Fixed a rare issue that could cause a **UM** application (including `umestored`) to crash on startup with `FATAL: failed assertion[info->umq_ack.flags&LBMC_UMQ_ACK_FLAG_TYPE_STABLE]`.

## 18.2.4. Known Issues

- Receivers using event queues and Spectrum with **UME** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]`. Changing `fd_management_type` (`../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 18.3. UMQ 5.0

### 18.3.1. Updates

- **Indexed Ultra Load balancing (ULB)** - Added indexing features to ULB, which is similar to indexed queuing and uses the same APIs (`lbm_rcv_umq_index_start_assignment()`, `lbm_rcv_umq_index_release()`, etc.), specifying the ULB source string in place of a queue name. See [Indexed Ultra Load Balancing \(ULB\)](#) (`../UME/ume-ops-operational-view.html#INDEXED-ULB`).
- Added the ability to set a message's total lifetime for both **UMQ** and ULB sources. See [Message Lifetimes and Reassignment](#) (`../UME/designing-queuing-applications.html#MESSAGE-LIFETIMES-REASSIGN`).

- **Local JMS Transaction Support - UM JMS** now supports local transaction.
- **JMS Application Header Attribute-** Added the `ConnectionFactory`. attribute, `USE_APP_HEADER`, allowing you to turn off the use of App Headers which can greatly increases performance. See `FactoryAttributes` (`../jms_concepts/index.html#FACTORYATTRIBUTES`).
- Removed the restriction requiring that the `umq_ulb_application_set` (`../Config/ultramessagingqueuingoptions.html#SOURCEUMQULBAPPLICATIONSET`) source option be specified prior to setting any other `umq_ulb_application_set_*` or `umq_ulb_receiver_*` options.

### 18.3.2. Bug Fixes

- Fixed an issue with ULB that prevented ULB receivers from completing registration when they attempted to register with a ULB source using an assignment ID (internal 32-bit identifier generated randomly at receiver startup) already in use by a different receiver at that source. This issue could cause the log warning: `received ACK with out-of-range appset index 65535/1`.
- Fixed a problem in the Java API (`../JavaAPI/html/index.html`) that caused a **UMP** store or **UMQ** queue to fail to acknowledge messages that are passed off to a different thread to be disposed.
- Added `LBM` as a `DEFAULT_TOPIC_TYPE` and `DEFAULT_TEMP_TOPIC_TYPE` in the JMS App Header. This allows Request/Reply to work without **UMP**.

## 19. Release LBM 4.2.8 / UME 3.2.8 / UMQ 2.1.8 - July 2011

### 19.1. LBM 4.2.8

#### 19.1.1. Updated Features

- Implemented a more efficient structure in the UM Gateway for detecting duplicate sequence numbers. This addition includes the new **29West** configuration options, `mim_sqn_window_increment` (`../Config/multicastimmediatemessagingoperationoptions.html#CONTEXTMIMSQNWINDWINCREMENT`) and `mim_sqn_window_size` (`../Config/multicastimmediatemessagingoperationoptions.html#CONTEXTMIMSQNWINDOWSIZE`), in addition to the UM Gateway configuration element, `<sqn-window size="16384" increment="8192"/>`. See **29West Gateway Configuration DTD** (`../Gateway/tnwgd-dtd-doc/index.html`).
- In the UM Gateway, added the topic name to both endpoint and peer portal log messages and the source string to endpoint portal log messages when a new transport is joined and a forwarding entry for the source's context instance (`ctxinst`) already exists with a different association. Log messages now appear in the following form: `endpoint portal [XX] found address for source ctxinst [YY] (topic [AA] source [BB]) different from existing association - new address used (rcvdc create)`, where `AA` is the topic string, and `BB` is the source string.

- Implemented backward compatibility for LBMMON statistics packets for LBM Version 3.6.1 and later by enhancing LBMMON internal message processing.

### 19.1.2. Bug Fixes

- Corrected a problem with the UM Gateway that resulted in a receiver experiencing unrecoverable loss while in a Late Join operation with a not fully initialized proxy source.
- Improved the efficiency of wildcard pattern interest messages between UM Gateway peer portals which previously could cause spikes in traffic and message latency.
- Corrected a problem with the UM Gateway that caused topic queries looking for a **UMS** context to continually loop.
- Corrected a problem with the UM Gateway that prevented source side filtering from working properly.
- Corrected a problem with a late joining receiver that sometimes caused extra, unnecessary late join requests to be sent to a source, resulting in spurious unrecoverable loss events on the receiver at the beginning of a late join.
- Corrected a problem that could cause late join through a UM Gateway to fail to catch up to the live stream when requested messages were no longer available at the original source.

### 19.1.3. Known Issues

- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: LOG Level 5:  
LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **UMS** may leave a monitoring thread running after context deletion. Informatica is investigating this problem.



- The UM Gateway does not currently support responses that exceed the maximum datagram size.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

## 19.2. UME 3.2.8

### 19.2.1. Bug Fixes

No specific **UMP** problems were discovered.

### 19.2.2. Known Issues

- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]. Changing fd_management_type` (`../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 19.3. UMQ 2.1.8

### 19.3.1. Bug Fixes

No specific UMQ problems were discovered.

## 20. Release LBM 4.2.7 / UME 3.2.7 / UMQ 2.1.7 - July 2011

### 20.1. LBM 4.2.7

#### 20.1.1. Bug Fixes

- Corrected a problem for 32-bit Java applications only, which resulted in the `getStores()` method of the `LBMSourceAttributes` class returning incorrect registration IDs or throwing an exception stating `Store entry registration ID was < 0`.
- Fixed an issue that could cause the logging of an `IllegalArgumentException` in Java applications when receiving responses.
- Fixed an issue with Late Join that caused a late joining receiver to send extra, unnecessary late join requests to a source, which resulted in spurious unrecoverable loss events on the receiver at the beginning of a late join.

#### 20.1.2. Known Issues

- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), UMS 4.1 issues the following warning: LOG Level 5: `LBT-RDMA: VMS Logger Message (Error): vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces,

which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.

- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, UMS may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- The UM Gateway does not currently support responses that exceed the maximum datagram size.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

## 20.2. UME 3.2.7

### 20.2.1. Bug Fixes

No specific UMP problems were discovered.

### 20.2.2. Known Issues

- Receivers using event queues and Spectrum with UMP can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]`. Changing `fd_management_type`

(../Config/majoroptions.html#CONTEXTFDMANAGEMENTTYPE) to `devpoll` prevents this problem. Informatica is investigating this problem.

- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 20.3. UMQ 2.1.7

### 20.3.1. Bug Fixes

No specific UMQ problems were discovered.

## 21. Release LBM 4.2.6 / UME 3.2.6 / UMQ 2.1.6 - June 2011

### 21.1. LBM 4.2.6

#### 21.1.1. Bug Fixes

- Single-TCP acceptor peer portals and TCP peer portals now work properly when using the UM Gateway on Microsoft Windows. This was listed as a Known Issue in UMS 4.2.5.
- Corrected an error involving how the UM Gateway handled ACL port comparisons.

#### 21.1.2. Known Issues

- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.

- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: LOG Level 5:  
LBT-RDMA: VMS Logger Message (Error): vmss\_create\_store: 196[E] vms\_listen:  
rdma\_bind\_addr failed (r=-1). This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **UMS** may leave a monitoring thread running after context deletion. Informatica is investigating this problem.
- The UM Gateway does not currently support responses that exceed the maximum datagram size.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).

## 21.2. UME 3.2.6

### 21.2.1. Bug Fixes

No specific **UMP** problems were discovered.

### 21.2.2. Known Issues

- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.

- 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]. Changing `fd_management_type` (`../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- UMP proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 21.3. UMQ 2.1.6

### 21.3.1. Bug Fixes

No specific UMQ problems were discovered.

## 22. Release LBM 4.2.5 / UME 3.2.5 / UMQ 2.1.5 - June 2011

### 22.1. LBM 4.2.5

#### 22.1.1. Updated Features

- Added support for intentional gaps in Hot Failover message streams. See Hot Failover Intentional Gap Support (`../Design/lbm-features.html#HF-Intentional-Gap`). Also added support for Hot Failover optional messages that HF receivers can be configured to receiver or not receive. See Hot Failover Optional Messages (`../Design/lbm-features.html#Optional-HF-Messages`).

#### 22.1.2. Bug Fixes

- Corrected a problem that could cause a crash of an *initiator* UM Gateway configured with a single-TCP peer connection if the *acceptor* gateway was not yet running.
- The configuration option `delivery_control_maximum_total_map_entries` (`../Config/deliverycontroloptions.html#CONTEXTDELIVERYCONTROLMAXIMUMTOTALMAPENTRIES`) was not being applied on a per thread basis when `use_transport_thread` (`../Config/majoroptions.html#RECEIVERUSETRANSPORTTHREAD`) was set as intended. This option is now applied correctly on a per thread basis.
- When setting the configuration options `delivery_control_loss_check_interval` (`../Config/deliverycontroloptions.html#RECEIVERDELIVERYCONTROLLOSSCHECKINTERVAL`) and `use_transport_thread` (`../Config/majoroptions.html#RECEIVERUSETRANSPORTTHREAD`) together, the

loss check timer started on the incorrect thread and subsequently produced the Fatal Assert `orderrec->msg!=NULL` or crash. This timer is now scheduled on the correct thread.

- Corrected a cross-version problem that could cause a fatal assert when an application deletes a **UMS 4.X** receiver that has received data from a **UMS 3.X** source.
- Added defensive measures in the UM Gateway to guard against the delayed delivery (tens of seconds) of internal events after they have been enqueued, possibly due to CPU starvation.
- Corrected a problem in the UM Gateway which could result in a deadlock when using a single-TCP peer connection (either acceptor or initiator).
- Corrected a format specifier in a debug log statement for LBT-RU sources which could lead to a segfault.
- Corrected a problem in the UM Gateway that prevented a topic matching a pattern from being immediately forwarded. This only occurred when there were at least 3 topic resolution domains, the source was accessible only via a peer connection from one domain, and wildcard receivers specifying overlapping but different patterns existed in the other two domains.

### 22.1.3. Known Issues

- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- Sending LBT-IPC messages larger than 65,535 bytes is not supported when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero) unless you set `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced`.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **UMS 4.1** issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **UMS** may leave a monitoring thread running after context deletion. This situation produces the following error: `VMA ERROR : epoll_wait_call:36:epoll_wait_call() epfd 48 not found`. Informatica is investigating this problem.
- The UM Gateway does not currently support responses that exceed the maximum datagram size.

- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- The UM Gateway does not currently support a four gateway "full-mesh" configuration (i.e., all gateways connected).
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- Informatica recommends against configuring UM Gateways in a ring of peer portals - use configurations utilizing gateway endpoints (parallel paths) to break the ring.
- Informatica recommends not stopping and restarting UM Gateways within the transport's activity timeout period (`transport_*_activity_timeout` defaults to 60 seconds for LBTRU and LBTRM).
- When using the UM Gateway on Microsoft Windows, single-TCP acceptor peer portals and TCP peer portals do not work. This issue will be corrected in a future release. Please contact us at <http://29west.com/support> for more information.

## 22.2. UME 3.2.5

### 22.2.1. Bug Fixes

No specific UMP problems were discovered.

### 22.2.2. Known Issues

- Receivers using event queues and Spectrum with UMP can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]. Changing fd_management_type` (`../Config/majoroptions.html#CONTEXTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- UMP proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.



## 22.3. UMQ 2.1.5

### 22.3.1. Bug Fixes

No specific UMQ problems were discovered.

## 23. Release LBM 4.2.4 / UME 3.2.4 / UMQ 2.1.4 - May 2011

### 23.1. LBM 4.2.4

#### 23.1.1. Bug Fixes

- 4117
- When using Multi-Transport Threads, a deadlock was possible if you deleted a receiver at the same moment a transport session began or ended. An alternative locking mechanism has been implemented to prevent a deadlock when you create new transport sessions or delete old transport sessions using the same transport thread as a deleted receiver. *Note:* This solution was erroneously cited as being made in the LBM 4.2.3 release. It is effective with the LBM 4.2.4 release.
- 4612
- 4623
- 4624
- 4625

#### 23.1.2. Known Issues

- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Multi-Transport Threads do not support persistent stores (**UMP**) or queues (**UMQ**).
- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- When using LBT-IPC, a seg fault can occur when sending messages larger than 65,535 bytes when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero). The seg fault occurs when fragments are lost. Setting `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced` avoids the seg fault by eliminating loss. Informatica is investigating this issue.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.

- If you use the current version of VMS (3.2.8), UMS 4.1 issues the following warning: LOG Level 5:  
LBT-RDMA: VMS Logger Message (Error): vmss\_create\_store: 196[E] vms\_listen:  
rdma\_bind\_addr failed (r=-1). This warning indicates that rdma\_bind failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts rdma\_bind on all interfaces. When released, VMS version 3.2.9 will only run rdma\_bind on infiniband-capable interfaces.
- When using Automatic Monitoring (../Config/automaticmonitoringoptions.html) with ud\_acceleration (../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION) and the epoll file descriptor option, UMS may leave a monitoring thread running after context deletion. This situation produces the following error: VMA ERROR : epoll\_wait\_call:36:epoll\_wait\_call() epfd 48 not found. Informatica is investigating this problem.
- A requesting application using a UM Gateway may not receive the expected number of responses. The problem occurs if the responses exceed the maximum datagram size, are fragmented and the gateway must forward simultaneous fragmented responses. In this case the fragments may become intermingled and result in a response not being delivered as expected. The requesting application receives a warning similar to: WARNING: failed assertion [offset==0] at line 1627 in ../../../../src/lib/lbm/lbmqr.c. Informatica is investigating this problem.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- When using the UM Gateway in a four gateway "full-mesh" configuration (i.e., all gateways connected) the following instabilities have been observed.
  - Requesting applications can receive multiple or unending responses.
  - Restarting a single gateway has resulted in the crash of other gateways.
  - Running the same set of tests simultaneously in 2 opposite directions over a set of gateways causes fatal asserts in the UM Gateways. Although this is not a typical user configuration, more complex gateway configurations could simulate this effect. The UM Gateway is designed to successfully forward the same set of topic messages sent both ways across a set of gateways.

These behaviors have not been seen in other, more typical gateway configurations. Informatica is investigating this problem.

- Under certain extreme conditions manufactured by configuring atypically low timeout settings, the UM Gateway has been observed to terminate with various fatal assertions. We believe this to be the result of a race condition exploited by the unusually low timer settings. This issue has not been seen with less aggressive or default timer settings. Informatica continues to investigate this problem.
- In some cases, a UM Gateway instance does not establish a reliable peer connection with another UM Gateway instance residing on the same Solaris 64-bit host. At this time, Informatica does not recommend configuring 1 or more peer-connected gateway instances on a single Solaris 64-bit host. Informatica continues to investigate this problem.
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.

- When using the UM Gateway in a failover configuration using LBT-RDMA, if the gateways experience multiple failures, receivers may experience deafness (unable to discover sources). For this to occur, all gateways must fail once and at least one gateway must fail more than once.
- Informatica has discovered that simultaneous event processing that involves EOS's and topic advertisements (TIR) combined with receiver deletes can cause an application to crash when using Multi-Transport Threads. Applications which routinely create and destroy receivers as part of normal running should disable Multi-Transport Threads to avoid this risk. Informatica is working on a solution.

## 23.2. UME 3.2.4

### 23.2.1. Bug Fixes

No specific **UMP** problems were discovered.

### 23.2.2. Known Issues

- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]`. Changing `fd_management_type` (`./Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 23.3. UMQ 2.1.4

### 23.3.1. Bug Fixes

No specific **UMQ** problems were discovered.

## 24. Release LBM 4.2.3 / UME 3.2.3 / UMQ 2.1.3 - April 2011

### 24.1. LBM 4.2.3

#### 24.1.1. Bug Fixes

- Corrected a problem that caused a fatal assert if a receiving application receives messages from a sending application that repeatedly creates and deletes sources on the same topic. This problem originated in the Microsoft Windows release of LBM 4.2.2. Previous versions did not have this problem.
- When using Multi-Transport Threads, a deadlock was possible if you deleted a receiver at the same moment a transport session began or ended. An alternative locking mechanism has been implemented to prevent a deadlock when you create new transport sessions or delete old transport sessions using the same transport thread as a deleted receiver.

#### 24.1.2. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- When using LBT-IPC, a seg fault can occur when sending messages larger than 65,535 bytes when `ordered_delivery` (`./Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero). The seg fault occurs when fragments are lost. Setting `transport_lbtipc_behavior` (`./Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced` avoids the seg fault by eliminating loss. Informatica is investigating this issue.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), UMS 4.1 issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): vmss\_create\_store: 196[E] vms\_listen: rdma\_bind\_addr failed (r=-1). This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`./Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`./Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, UMS may leave a monitoring thread running after context deletion. This situation produces the following error: VMA ERROR : `epoll_wait_call:36:epoll_wait_call() epfd 48 not found`. Informatica is investigating this problem.
- A requesting application using a UM Gateway may not receive the expected number of responses. The problem occurs if the responses exceed the maximum datagram size, are fragmented and the gateway must forward simultaneous fragmented responses. In this case the fragments may become intermingled and result in a response not being delivered as expected. The requesting application receives a warning similar to: WARNING: `failed assertion [offset==0] at line 1627 in ../../../../src/lib/lbm/lbmqr.c`. Informatica is investigating this problem.

- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- When using the UM Gateway in a four gateway "full-mesh" configuration (i.e., all gateways connected) the following instabilities have been observed.
  - Requesting applications can receive multiple or unending responses.
  - Restarting a single gateway has resulted in the crash of other gateways.
  - Running the same set of tests simultaneously in 2 opposite directions over a set of gateways causes fatal asserts in the UM Gateways. Although this is not a typical user configuration, more complex gateway configurations could simulate this effect. The UM Gateway is designed to successfully forward the same set of topic messages sent both ways across a set of gateways.

These behaviors have not been seen in other, more typical gateway configurations. Informatica is investigating this problem.

- Under certain extreme conditions manufactured by configuring atypically low timeout settings, the UM Gateway has been observed to terminate with various fatal assertions. We believe this to be the result of a race condition exploited by the unusually low timer settings. This issue has not been seen with less aggressive or default timer settings. Informatica continues to investigate this problem.
- In some cases, a UM Gateway instance does not establish a reliable peer connection with another UM Gateway instance residing on the same Solaris 64-bit host. At this time, Informatica does not recommend configuring 1 or more peer-connected gateway instances on a single Solaris 64-bit host. Informatica continues to investigate this problem.
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- When using the UM Gateway in a failover configuration using LBT-RDMA, if the gateways experience multiple failures, receivers may experience deafness (unable to discover sources). For this to occur, all gateways must fail once and at least one gateway must fail more than once.
- When using the UM Gateway Informatica recommends that you **do not** enable the Multi-Transport Threads feature. Informatica does not support nor test the operation of Multi-Transport Threads across the UM Gateway.
- Informatica has discovered that simultaneous event processing that involves EOS's and topic advertisements (TIR) combined with receiver deletes can cause an application to crash when using Multi-Transport Threads. Applications which routinely create and destroy receivers as part of normal running should disable Multi-Transport Threads to avoid this risk. Informatica is working on a solution.

## 24.2. UME 3.2.3

### 24.2.1. Bug Fixes

- Fixed a problem with source failover to proxy sources that resulted in the following warning: `NOTICE: store [ip:port] reports it has not received TIR. Possible misconfiguration?`
- Fixed a problem that allowed **UMP** proxy source election notifications to loop forever through UM Gateways configured in a loop.
- Fixed a problem that could cause a deadlock when message acknowledgements were sent at the same time that a transport session was closed.

### 24.2.2. Known Issues

- Receivers using event queues and Spectrum with **UMP** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]. Changing fd_management_type (./Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE) to devpoll prevents this problem.` Informatica is investigating this problem.
- **UMP** proxy source elections can only occur among stores in the same topic resolution domain connected directly via peer links. Informatica is investigating this problem.

## 24.3. UMQ 2.1.3

### 24.3.1. Bug Fixes

No specific UMQ problems were discovered.

## 25. Release LBM 4.2.2 / UME 3.2.2 / UMQ 2.1.2 - April 2011

### 25.1. LBM 4.2.2

#### 25.1.1. Bug Fixes

- In the UM Gateway, log messages that appear upon restart indicating a `forwarding entry` or `ctxinst` have been changed from WARNING to INFORMATIONAL. These log messages are normal and do not appear as the gateway discovers the surrounding environment and sets up the required forwarding information.
- Resolved a known issue with the UM Gateway that was reported in **LBM 4.2.1 *Known Issues***. Previously if you brought down a UM Gateway, reconfigured its number and/or type of portals and then restarted it, you had to also restart any other gateways with which that reconfigured gateway either connected to previously or currently connects to. Now you do not have to restart any connecting gateways.
- Fixed a problem with the Java API (`../JavaAPI/html/index.html`) that could cause a crash of the JVM if a user's source event callback threw an uncaught exception.
- Corrected a problem that prevented LBM 4.0 receivers from joining multiple pre-LBM 4.0 sources sending on the same topic if the topic advertisements for those sources were sent before the receivers were created.

#### 25.1.2. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- When using LBT-IPC, a seg fault can occur when sending messages larger than 65,535 bytes when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero). The seg fault occurs when fragments are lost. Setting `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced` avoids the seg fault by eliminating loss. Informatica is investigating this issue.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **LBM 4.1** issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **LBM** may leave a monitoring thread running after context deletion. This situation produces the following error: `VMA ERROR : epoll_wait_call:36:epoll_wait_call() epfd 48 not found`. Informatica is investigating this problem.
- A requesting application using a UM Gateway may not receive the expected number of responses. The problem occurs if the responses exceed the maximum datagram size, are fragmented and the gateway must forward

simultaneous fragmented responses. In this case the fragments may become intermingled and result in a response not being delivered as expected. The requesting application receives a warning similar to: `WARNING: failed assertion [offset==0] at line 1627 in ../../../../src/lib/lbm/lbmqr.c`. Informatica is investigating this problem.

- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- When using the UM Gateway in a four gateway "full-mesh" configuration (i.e., all gateways connected) the following instabilities have been observed.
  - Requesting applications can receive multiple or unending responses.
  - Restarting a single gateway has resulted in the crash of other gateways.
  - Running the same set of tests simultaneously in 2 opposite directions over a set of gateways causes fatal asserts in the UM Gateways. Although this is not a typical user configuration, more complex gateway configurations could simulate this effect. The UM Gateway is designed to successfully forward the same set of topic messages sent both ways across a set of gateways.

These behaviors have not been seen in other, more typical gateway configurations. Informatica is investigating this problem.

- Under certain extreme conditions manufactured by configuring atypically low timeout settings, the UM Gateway has been observed to terminate with various fatal assertions. We believe this to be the result of a race condition exploited by the unusually low timer settings. This issue has not been seen with less aggressive or default timer settings. Informatica continues to investigate this problem.
- In some cases, a UM Gateway instance does not establish a reliable peer connection with another UM Gateway instance residing on the same Solaris 64-bit host. At this time, Informatica does not recommend configuring 1 or more peer-connected gateway instances on a single Solaris 64-bit host. Informatica continues to investigate this problem.
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- When using the UM Gateway in a failover configuration using LBT-RDMA, if the gateways experience multiple failures, receivers may experience deafness (unable to discover sources). For this to occur, all gateways must fail once and at least one gateway must fail more than once.

## 25.2. UME 3.2.2

### 25.2.1. Bug Fixes

- Added the `-t` option to specify store names in the .NET and Java version of `umesrc`.



## 25.2.2. Known Issues

- Receivers using event queues and Spectrum with **UME** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]`. Changing `fd_management_type` (`./Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) to `devpoll` prevents this problem. Informatica is investigating this problem.

## 25.3. UMQ 2.1.2

### 25.3.1. Bug Fixes

- Fixed a problem that occurred when using either the Java API (`./JavaAPI/html/index.html`) or the .NET API (`./DotNetAPI/doc/Index.html`) and a source that sends using a per-send client object. The problem resulted in a crash or null pointer exception if the source was configured with both ULB and UME settings. If configured with only ULB settings, a memory leak of the per-send client object occurred. The following shows how you might send messages using `LBMContext.send()` in Java or .NET with a per-send client object.

```
LBMSourceSendExInfo exinfo = new LBMSourceSendExInfo();
exinfo.setClientObject(something);
context.send( . . . . , exinfo);
```

## 26. Release LBM 4.2.1 / UME 3.2.1 / UMQ 2.1.1 - April 2011

### 26.1. LBM 4.2.1

#### 26.1.1. New Features

- Added new configuration option, `resolver_advertisement_send_immediate_response` (`./Config/resolveroperationoptions.html#SOURCERESOLVERADVERTISEMENTSSENDIMMEDIATERESPONSE`) that allows you to disable the normal immediate response to queries and wildcard queries.

## 26.1.2. Updated Features

- Added the topic resolution request function to the Java API (`../JavaAPI/html/index.html`) and the .NET API (`../DotNetAPI/doc/Index.html`). Contexts may now request topic resolution through the `LBMContext.requestTopicResolution()` API. Also added the sample application, `lbmtrreq` to the Java and .NET collections of sample applications.

## 26.1.3. Bug Fixes

- Corrected numerous problems with the stability and reliability of the UM Gateway.
- In the UM Gateway, fixed a small memory leak that occurred when a requester processed large fragmented responses.
- In the UM Gateway, fixed an issue that could in specific circumstances cause a receiver downstream of a restarted gateway to become deaf to some remote sources.
- Corrected a problem with the UM Gateway that occurred when a gateway rejected a topic and then accepted it. This produced the assertion, `[warning] WARNING: failed assertion [topic->known_srcs==0]` at line 2597 in `../../../../src/lib/lbm/lbmresolve.c`.
- Resolved an issue that caused errors in Java during source event callbacks to corrupt TCP streams.

## 26.1.4. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.
- When using LBT-IPC, a seg fault can occur when sending messages larger than 65,535 bytes when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDELIVERY`) has been set to 0 (zero). The seg fault occurs when fragments are lost. Setting `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced` avoids the seg fault by eliminating loss. Informatica is investigating this issue.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **LBM 4.1** issues the following warning: LOG Level 5: `LBT-RDMA: VMS Logger Message (Error): vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **LBM** may leave a monitoring thread running after context deletion. This situation produces the following error: `VMA ERROR : epoll_wait_call:36:epoll_wait_call() epfd 48 not found`. Informatica is investigating this problem.

- A requesting application using a UM Gateway may not receive the expected number of responses. The problem occurs if the responses exceed the maximum datagram size, are fragmented and the gateway must forward simultaneous fragmented responses. In this case the fragments may become intermingled and result in a response not being delivered as expected. The requesting application receives a warning similar to: `WARNING: failed assertion [offset==0] at line 1627 in ../../../../src/lib/lbm/lbmqr.c`. Informatica is investigating this problem.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- When using the UM Gateway in a four gateway "full-mesh" configuration (i.e., all gateways connected) the following instabilities have been observed.
  - Requesting applications can receive multiple or unending responses.
  - Restarting a single gateway has resulted in the crash of other gateways.
  - Running the same set of tests simultaneously in 2 opposite directions over a set of gateways causes fatal asserts in the UM Gateways. Although this is not a typical user configuration, more complex gateway configurations could simulate this effect. The UM Gateway is designed to successfully forward the same set of topic messages sent both ways across a set of gateways.

These behaviors have not been seen in other, more typical gateway configurations. Informatica is investigating this problem.

- Under certain extreme conditions manufactured by configuring atypically low timeout settings, the UM Gateway has been observed to terminate with various fatal assertions. We believe this to be the result of a race condition exploited by the unusually low timer settings. This issue has not been seen with less aggressive or default timer settings. Informatica continues to investigate this problem.
- In some cases, a UM Gateway instance does not establish a reliable peer connection with another UM Gateway instance residing on the same Solaris 64-bit host. At this time, Informatica does not recommend configuring 1 or more peer-connected gateway instances on a single Solaris 64-bit host. Informatica continues to investigate this problem.
- If using LBT-RDMA across the UM Gateway and you exit the gateway with **Ctrl-C**, you may see a segfault. Informatica is aware of this and has not observed any ill effects from this segfault.
- If you bring down a UM Gateway, reconfigure its number and/or type of portals and then restart the gateway, you must also restart any other gateways with which that reconfigured gateway either connected to previously or now connects to. Informatica is investigating this issue.
- When using the UM Gateway in a failover configuration using LBT-RDMA, if the gateways experience multiple failures, receivers may experience deafness (unable to discover sources). For this to occur, all gateways must fail once and at least one gateway must fail more than once.

## 26.2. UME 3.2.1

### 26.2.1. Bug Fixes

- Fixed an issue where a UME source reached a state where it could not successfully register with UME stores located on the far side of one or more gateways. This situation only occurred if there were multiple paths from the source to the stores, the paths involved one or more peer links, and the gateways were brought down and up in rapid succession.
- Fixed an issue where messages could be prematurely reclaimed if reclamation is forced after losing registration.
- Fixed a condition that could cause a UME store to become effectively deaf to new data after a gateway failover and restart.

### 26.2.2. Known Issues

- Receivers using event queues and Spectrum with **UME** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.
- 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]. Changing fd_management_type (./Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE) to devpoll` prevents this problem. Informatica is investigating this problem.

## 26.3. UMQ 2.1.1

### 26.3.1. Bug Fixes

- Fixed an issue that could cause Java ULB and **UMQ** receivers to crash when explicitly reassigning a message.
- Fixed a problem that prevented the `reassigned` flag from being set for reassigned **UMQ** Serial Queue Dissemination (SQD) messages.

## 27. Release LBM 4.2 / UME 3.2 / UMQ 2.1 - March 2011

### 27.1. LBM 4.2

#### 27.1.1. New Features

- Added new receiver scope configuration option, `transport_demux_tablesz` (`../Config/majoroptions.html#RECEIVERTRANSPORTDEMUXTABLESZ`) that sets the size of the table used for mapping topic indices to delivery controllers.
- Added new configuration option, `network_compatibility_mode` (`../Config/majoroptions.html#CONTEXTNETWORKCOMPATIBILITYMODE`) to restrict the types of message headers that can be sent onto the network, for the purpose of improved backward compatibility.
- Added new configuration option, `retransmit_initial_sequence_number_request` (`../Config/latejoinoptions.html#RECEIVERRETRANSMITINITIALSEQUENCENUMBERREQUEST`) that allows topic advertisements (TIR) to trigger a late join if receivers detect a gap in messages instead of waiting for either a subsequent message or a TSNI.
- Added `lbm_send_request_ex()` call to the **LBM C** API (with corresponding `LBMSource.send()` methods in the Java and .NET APIs) to allow sending requests with `lbm_src_send_ex_info_t` or `LBMSourceSendExInfo` objects and per-send client objects.
- Added a new environment variable, `LBM_DEBUG_MAXSIZE`, that allows the LBM DEBUG log to roll over when the DEBUG log exceeds the maximum size specified by this variable (in bytes). The log rolls over slightly before reaching the limit, because adding the current log message to the log would exceed the maximum set with the environment variable.

#### 27.1.2. Updated Features

- Added the Hot Failover Concentrator (HFX) feature that allows Hot Failover streams on multiple contexts to be treated as a single hot failover stream. See [Hot Failover Across Multiple Contexts](#) (`../Design/lbm-features.html#HFX`).
- TSNI (transport topic sequence number information) have been enabled by default on TCP transports. See also `transport_topic_sequence_number_info_interval` (`../Config/majoroptions.html#SOURCETRANSPORTTOPICSEQUENCENUMBERINFOINTERVAL`).
- Modified internal timer handling to be more judicious when waking up the context thread to schedule a timer.
- Added command line delay option in seconds (`-d, --delay=NUM`) for C, Java and .NET sample applications that send messages.
- Implemented batching in communication between UM Gateway peer portals to improve throughput.
- The UM Gateway now allows empty (zero-length) topics and wildcard topic patterns.
- Context topic advertisements may now be disabled by setting, `resolver_context_advertisement_interval`

(../Config/majoroptions.html#CONTEXTRESOLVERCONTEXTADVERTISEMENTINTERVAL) to 0 (zero).

**Note:** Disabling context advertisements is not recommended unless specifically advised to do so.

- Zero Object Delivery (ZOD) is now enabled by default for both Java and .NET. Applications wishing to take advantage of ZOD should call `Dispose()` on messages before the end of the message callback.

**Addendum:** To clarify, please note that ZOD is now an always-on feature whenever you call `Dispose()`. If you do not call `Dispose()`, you must call `Promote()` to promote the message to an object. Conversely, if you do not call `Promote()`, you must call `Dispose()`.

- Added the new configuration option `mim_delivery_control_loss_check_interval` (../Config/deliverycontroloptions.html#CONTEXTMIMDELIVERYCONTROLLOSSCHECKINTERVAL) to periodically check Multicast Immediate Messaging (MIM) delivery controllers for loss. Option defaults to 0 and follows the same basic semantics as `delivery_control_loss_check_interval` (../Config/deliverycontroloptions.html#RECEIVERDELIVERYCONTROLLOSSCHECKINTERVAL).
- Application linking with the SmartHeap memory manager is no longer permitted under any provision of an **LBM** license alone. Existing Informatica applications that use the SmartHeap allocator, such as the **UME** store daemon (`umestored`) and the **UMCache** daemon (`umcached`) will continue to use the SmartHeap allocator.

### 27.1.3. Bug Fixes

- Added an `isClosed()` function to `LBMSource.cs` and `LBMSource.java` to check if a source object has a valid C reference.
- Fixed a Fatal Assert caused by a race condition that could occur when deleting an LBT-RDMA receiver at the same moment a source disconnect event was being processed.
- Corrected a problem with LBT-RU and LBT-RDMA that caused receivers to bind to the wrong source address. These transports now check the `IN_ADDRANY` value before overwriting the source IP address.
- Corrected a problem when running a context in sequential mode that prevented the transport threads from creating the event processing thread. Transport threads now force embedded mode.
- Fixed a problem that prevented non-blocking sends from returning `EWOULDBLOCK` when either the rate limiter was met or the inflight messages met the flight size.
- Corrected a UM Gateway issue that caused the fatal assertion, `FATAL: failed assertion [wt->topic!=NULL]` at line 4444 in `../src/gateway/tnwgendpoint.c`.
- Fixed a problem on Solaris i386 that caused TCP sends to always be blocking sends.
- Corrected misspellings on the UM Gateway peer portal Web Monitor page.
- Corrected several UM Gateway Web Monitor statistics that were being updated incorrectly, resulting in values which were too large. The statistics involved are:

On the main page:

- Unknown transport fragment bytes received

Endpoint sending:

- Transport topic bytes dropped due to no source
- Duplicate transport topic bytes dropped

Endpoint receiving:

- Transport topic bytes received from an unknown source
  - Transport topic request bytes received with no forwarding info
  - Immediate topic bytes received for an unknown topic
  - Immediate topic bytes received
  - Transport topic bytes received
  - Immediate topicless bytes received
  - Unicast data bytes received
- Fixed a seg fault that could occur when canceling a recurring timer from within the callback of that recurring timer.
  - Added the ability to forward (flood) Multicast Immediate Messaging (MIM) traffic from a UM Gateway portal. You enable this feature by configuring the portal with the `<flood-mim/>` element in the UM Gateway configuration file.
  - Fixed a problem in the UM Gateway that prevented topic Multicast Immediate Messaging (MIM) messages for which no receiver exists at the endpoint portal from updating the endpoint's MIM statistics. The endpoint now accurately reports the number of topic MIM messages that have been dropped due to no receiver for the topic.
  - Corrected a problem with the Multi-Transport Threads feature that consisted of the control block managing the thread pool per context never being freed. It is now freed upon context deletion.
  - Corrected two small memory leaks in the UM Gateway.
  - Corrected a problem with Multicast Immediate Messaging (MIM) that resulted in the callback specified by `immediate_message_receiver_function(..Config/multicastimmediatemessagingoperationoptions.html#CONTEXTIMMEDIATEMESSAGERECEIVERFUNCTION)` not to be called if a wildcard receiver existed in the context.
  - Removed the statistics, `Transport topic loss/loss burst events received` and `Immediate loss events received` from the UM Gateway Web Monitor for Endpoint Portals. Since the UM Gateway uses arrival order delivery, no loss events are ever delivered, so these statistics would always be zero.
  - An issue was corrected in the UM Gateway which could lead to the non-fatal assert, `WARNING: failed assertion [result!=0] at line 416 in ../../../../src/lib/lbm/lbmrxsctrl.c`. This could also lead to another fatal assert, `FATAL: failed assertion [info->sqn==ctrl->trail_sqn] at line 249 in ../../../../src/lib/lbm/lbmrxsctrl.c`
  - Fixed a problem that could cause some of multiple threads calling `lbm_send_response()` to hang indefinitely.
  - Corrected numerous problems with the stability and reliability of the UM Gateway.
  - Fixed a problem that could sometimes cause `lbm_context_topic_resolution_request()` to incorrectly return failure when it had actually succeeded.

#### 27.1.4. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. Informatica is investigating this issue.

- When using LBT-IPC, a seg fault can occur when sending messages larger than 65,535 bytes when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero). The seg fault occurs when fragments are lost. Setting `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced` avoids the seg fault by eliminating loss. Informatica is investigating this issue.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. Informatica is investigating this problem.
- If you use the current version of VMS (3.2.8), **LBM 4.1** issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **LBM** may leave a monitoring thread running after context deletion. This situation produces the following error: `VMA ERROR : epoll_wait_call:36:epoll_wait_call() epfd 48 not found`. Informatica is investigating this problem.
- A requesting application using a UM Gateway may not receive the expected number of responses. The problem occurs if the responses exceed the maximum datagram size, are fragmented and the gateway must forward simultaneous fragmented responses. In this case the fragments may become intermingled and result in a response not being delivered as expected. The requesting application receives a warning similar to: `WARNING: failed assertion [offset==0] at line 1627 in ../../../../src/lib/lbm/lbmqr.c`. Informatica is investigating this problem.
- At the present time, 32-bit applications cannot interact with 64-bit applications using the LBT-IPC transport. As a result, a 64-bit UM Gateway cannot interact with a 32-bit application using LBT-IPC. It can only interact with a 64-bit application. Likewise, a 32-bit UM Gateway can only interact with a 32-bit application.
- The UM Gateway does not currently support queuing at this time, only streaming and persistence. This will be resolved in a future release.
- The UM Gateway is not supported as a standalone Windows service at this time. This will be resolved in a future release.
- When using the UM Gateway in a four gateway "full-mesh" configuration (i.e., all gateways connected) the following instabilities have been observed.
  - Requesting applications can receive multiple or unending responses.
  - Restarting a single gateway has resulted in the crash of other gateways.
  - Running the same set of tests simultaneously in 2 opposite directions over a set of gateways causes fatal asserts in the UM Gateways. Although this is not a typical user configuration, more complex gateway configurations could simulate this effect. The UM Gateway is designed to successfully forward the same set of topic messages sent both ways across a set of gateways.

These behaviors have not been seen in other, more typical gateway configurations. Informatica is investigating this problem.

- Under certain extreme conditions manufactured by configuring atypically low timeout settings, the UM Gateway has been observed to terminate with various fatal assertions. We believe this to be the result of a race condition



exploited by the unusually low timer settings. This issue has not been seen with less aggressive or default timer settings. Informatica continues to investigate this problem.

### 27.1.5. LBM 4.0 Changes

The following changes were implemented in **LBM** Release 4.0 but omitted from the **LBM 4.0 Updated Features** section. They have also been added to that section.

- Added ability to set a **LBM** Configuration Option interface by name. When setting by name the configuration option's value must be wrapped in quotes.

## 27.2. UME 3.2

### 27.2.1. New Features

- Added new source scope configuration option, `use_extended_reclaim_notifications` (`../Config/ultramessagingpersistenceoptions.html#SOURCEUMEUSEEXTENDEDRECLAIMNOTIFICATIONS`) that allows your application to receive the new, expanded message reclaim notification or the standard notification. The new notification or source event, `LBM_SRC_EVENT_UME_MESSAGE_RECLAIMED_EX`, contains a new field, `LBM_SRC_EVENT_UME_MESSAGE_RECLAIMED_EX_FLAG_FORCED` that **UME** sets if the reclamation is a forced reclaim.
- Added new configuration option, `ume_session_id` (`../Config/ultramessagingpersistenceoptions.html#CONTEXTUMESESSID`) that specifies a Session ID for a source, receiver or a context. A Session ID is a 64-bit value that uniquely identifies any set of sources with unique topics and receivers with unique topics. A single Session ID allows **UME** stores to correctly identify all the sources and receivers for a particular application. See also `Managing RegIDs with Session IDs` (`../UME/designing-persistent-applications.html#SESSION-IDS`).

### 27.2.2. Bug Fixes

- Fixed a problem that could cause a fatal assert in a **UME** store in a quorum/consensus configuration that uses a low `source-activity-timeout` setting when a source is rapidly added and then deleted.
- Fixed an issue with memory stores that could cause messages received by a store via late join to be discarded when the source starts.
- Fixed a problem that allowed **UME** stores with `allow-proxy-source` set to 0 (Disable) to participate in proxy source elections.
- Fixed a problem that causes **UME** receivers to report unrecoverable loss during recovery for messages that were unavailable at some stores, but still available at others.
- Corrected an issue where **UME** PREG error responses were not being forwarded through the gateway.

### 27.2.3. Known Issues

- Receivers using event queues and Spectrum with **UME** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.

## 27.3. UMQ 2.1

### 27.3.1. New Features

- Implemented Indexed Queuing which allows sources to send messages with an index (an unsigned 64-bit integer or a string up to 216 characters in length). The Queue assigns all messages sent with a particular index to an individual receiver in each Application Set. You can set policies for index assignment in the Queue's umstored XML configuration file. See Indexed Queuing ([../UME/umeconcepts.html#INDEXED-QUEUING](#)).
- Implemented a Dead Letter Queue feature to isolate unconsumed messages which prevents these messages from causing application or queuing system problems. See Dead Letter Queue ([../UME/umeconcepts.html#DEAD-LETTER-QUEUE](#)).
- Added the ability to set a message's total lifetime for both **UMQ** and ULB sources. See Message Lifetimes ([../UME/umeconcepts.html#MESSAGE-LIFETIMES](#)).
- Added `lbm_msg_umq_reassign()` in the C API ([../API/index.html](#)) to allow a **UMQ** or ULB receiver to either reassign or discard a message. Corresponding methods have been added in the Java API ([../JavaAPI/html/index.html](#)) and the .NET API ([../DotNetAPI/doc/Index.html](#)).
- Added two new receiver configuration options, `umq_ulb_source_check_interval` ([../Config/ultramessagingqueuingoptions.html#RECEIVERUMQULBSOURCECHECKINTERVAL](#)) and `umq_ulb_source_activity_timeout` ([../Config/ultramessagingqueuingoptions.html#RECEIVERUMQULBSOURCEACTIVITYTIMEOUT](#)) to allow a ULB receiver to proactively attempt re-registration with a ULB source if the receiver has not seen any activity (including keepalives) from that source in a specified amount of time, but the source's transport session is still alive and valid.

### 27.3.2. Bug Fixes

- Fixed a problem that caused a Queue using Serial Queue Dissemination (SQD) to incorrectly flag certain messages sent to receivers as IMMEDIATE.
- Fixed a problem that could cause message data corruption when a Queue received large, fragmented responses or late join messages with application header data.
- Fixed a problem that could prevent a receiver from requesting the retransmission of assigned messages from the Queue.

- Fixed a potential bus error on Solaris SPARC platforms that could occur upon the restart of a Queue's `umestored` daemon.

## 28. Release LBM 4.1.3 / UME 3.1.3 / UMQ 2.0.1 - November 2010

### 28.1. LBM 4.1.3

#### 28.1.1. Bug Fixes

- Corrected a problem in the UM Gateway that prevented proper Request/Response behavior.
- Modified the example request programs (`lbmreq`, `lbmmreq`, and `lbmireq`) to determine the request port in a more consistent manner.

#### 28.1.2. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. **29West** is investigating this issue.
- When using LBT-IPC, a seg fault can occur when sending messages larger than 65,535 bytes when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero). The seg fault occurs when fragments are lost. Setting `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced` avoids the seg fault by eliminating loss. **29West** is investigating this issue.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. **29West** is investigating this problem.
- If you use the current version of VMS (3.2.8), **LBM 4.1** issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **LBM** may leave a monitoring thread running after context deletion. This situation produces the following error: `VMA ERROR : epoll_wait_call:36:epoll_wait_call() epfd 48 not found`. **29West** is investigating this problem.

## 28.2. UME 3.1.3

### 28.2.1. Bug Fixes

No UME specific problems were identified.

### 28.2.2. Known Issues

- Receivers using event queues and Spectrum with UME can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use LBM\_EVQ\_BLOCK when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.

## 28.3. UMQ 2.0

### 28.3.1. Bug Fixes

No UMQ specific problems were identified.

## 29. Release LBM 4.1.2 / UME 3.1.2 / UMQ 2.0 - November 2010

### 29.1. LBM 4.1.2

#### 29.1.1. Bug Fixes

- Changed the `resolver_unicast_interface` (`../Config/unicastresolvernetworkoptions.html#CONTEXTRESOLVERUNICASTINTERFACE`) option to accept either multicast-capable or non-multicast interfaces.
- Corrected a UM Gateway issue that could cause a peer portal to become deaf to a wildcard pattern, if the peer portal received a wildcard receiver topic query (WC-TQR) for the pattern before the companion portal was connected.
- Corrected a problem with the UM Gateway that prevented topics from being purged when no receivers existed for them. This was evident in the Web Monitor's Gateway page that showed more active topics than any of the individual portals.

- Corrected a problem with `epoll` initialization that prevented `epoll` from being used with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`).
- Corrected a problem with the UM Gateway that created a deadlock if multiple, newly created sources immediately begin sending to a peer gateway.
- Corrected a segfault in a UM Gateway peer portal that might occur when a wildcard receiver has been deleted and the peer portal cancels interest in the deleted wildcard receiver's topic pattern.

### 29.1.2. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. **29West** is investigating this issue.
- When using LBT-IPC, a seg fault can occur when sending messages larger than 65,535 bytes when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero). The seg fault occurs when fragments are lost. Setting `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCESTRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced` avoids the seg fault by eliminating loss. **29West** is investigating this issue.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. **29West** is investigating this problem.
- If you use the current version of VMS (3.2.8), **LBM 4.1** issues the following warning: LOG Level 5:  
LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.
- When using Automatic Monitoring (`../Config/automaticmonitoringoptions.html`) with `ud_acceleration` (`../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION`) and the `epoll` file descriptor option, **LBM** may leave a monitoring thread running after context deletion. This situation produces the following error: `VMA ERROR : epoll_wait_call:36:epoll_wait_call() epfd 48 not found`. **29West** is investigating this problem.

### 29.1.3. LBM 4.0 Changes

The following changes were implemented in **LBM** Release 4.0 but omitted from the **LBM 4.0 Updated Features** section. They have also been added to that section.

- Blocking sends from within a context-thread callback are no longer permitted and now return an `LBM_EINVAL` error. This restriction eliminates a possible deadlock. Non-blocking sends from a context-thread callback are still permitted, and any type of send is permitted outside a context-thread callback.
- Some API calls that were formerly never safe to call from within a context-thread callback (due to a deadlock condition) are now always safe to call from within a context-thread callback. These API calls include:
  - `lbm_request_delete()`
  - `lbm_request_delete_ex()`
  - `lbm_response_delete()`

Additionally, the following send-related API calls are now safe to call from within a context-thread callback provided you set the LBM\_SRC\_NONBLOCK flag.

- `lbm_multicast_immediate_message()`
  - `lbm_multicast_immediate_request()`
  - `lbm_unicast_immediate_message()`
  - `lbm_unicast_immediate_request()`
  - `lbm_send_response()`
- Added ability to set a **LBM** Configuration Option interface by name. When setting by name the configuration option's value must be wrapped in quotes.

## 29.2. UME 3.1.2

### 29.2.1. New Features

- Added the following **UME** configuration options.
  - (source) `ume_activity_timeout`  
(../Config/ultramessagingpersistenceoptions.html#SOURCEUMEACTIVITYTIMEOUT)
  - (receiver) `ume_activity_timeout`  
(../Config/ultramessagingpersistenceoptions.html#RECEIVERUMEACTIVITYTIMEOUT)
  - (source) `ume_state_lifetime`  
(../Config/ultramessagingpersistenceoptions.html#SOURCEUMESTATELIFETIME)
  - (receiver) `ume_state_lifetime`  
(../Config/ultramessagingpersistenceoptions.html#RECEIVERUMESTATELIFETIME)
- Also added new new topic `ume-attributes` options, `source-state-lifetime` and `receiver-state-lifetime`. See `Options for a Topic's ume-attributes Element` (../UME/umestored-config.html#UMESTORED-TOPIC-OPTIONS).

The activity and state lifetime timers protect the Reg IDs of inactive sources and receivers and also controls how long **UME** retains the state and cache for inactive sources and receivers. These options can operate in conjunction with the proxy source option or independently. See `Activity Timeout and State Lifetime Options` (../UME/ume-fault-tolerance.html#ACTIVITY-TIMEOUT-STATE-LIFETIME) for a discussion of these timers work together and how they work with proxy sources.

### 29.2.2. Bug Fixes

- Fixed an issue that could result in a crash from leakage of a proxy source object in the store leading to a "double free" of the proxy source object.

- Fixed a problem that could cause **UME** receivers to crash if their source goes down and comes back repeatedly with the same reg ID and also comes back registered to more stores than previously.

### 29.2.3. Known Issues

- Receivers using event queues and Spectrum with **UME** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.

## 29.3. UMQ 2.0

### 29.3.1. New Features

- **JMS Support:** Added support for JMS with the **Ultra Messaging JMS**. See Ultra Messaging JMS API Quick Start ([../QuickStart/jms-binary-quick-start.html](#)) for information on running the example JMS client applications. See Ultra Messaging JMS API Guide ([../jms\\_concepts/index.html](#)) for conceptual, operational and configuration information about Ultra Messaging JMS.

### 29.3.2. Bug Fixes

- Corrected a problem that could cause a fatal assert if a queue received a MIM message not meant to be queued on a topic that exists in the queue due to other messages that were meant to be enqueued.

## 30. Release UME 3.0.2 - November 2010

### 30.1. UME 3.0.2

#### 30.1.1. Bug Fixes

- Fixed a problem that could cause **UME** receivers to crash if their source goes down and comes back repeatedly with the same reg ID and also comes back registered to more stores than previously.

## 31. Release LBM 4.1.1 / UME 3.1.1 / UMQ 1.1.1 - October 2010

### 31.1. LBM 4.1.1

#### 31.1.1. Updated Features

- Improved the efficiency of Late Join retransmissions with the addition of a timer for every request set to the same value as the `retransmit_request_interval` (`./Config/latejoinoptions.html#RECEIVERRETRANSMITREQUESTINTERVAL`). Previously, Late Join re-requested retransmissions if the retransmission was not received at the expiration of the `retransmit_request_interval`. Now Late Join only re-requests a retransmission when the first request's `retransmit_request_interval` expires. This change also applies to retransmission requests from a **UME** store.
- The UM Gateway now supports single connection TCP Peer Portals.
- Added the `<nodelay/>` element to the Gateway configuration for peer portals. This sets the `TCP_NODELAY` option for the peer TCP connections, disabling Nagle's algorithm. By default, `TCP_NODELAY` is not set.

#### 31.1.2. Bug Fixes

- Fixed an issue that could cause some source events for Hot Failover sources in the .NET API to not be delivered to the source's source event callback for a brief period of time after source creation.
- Corrected a problem with LBT-IPC timers that caused the log warning: `THROTTLED MSG: timer scheduled <= MIN_CLOCK_RES_MSEC (2 ms) [1282330517.878809]: Rescheduling for 3 ms.` This problem occurred if `transport_lbtipc_behavior` (`./Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) was set to `receiver_paced` and the source was slowed by a receiver that could not keep up to the data stream.
- Corrected an issue with `lbmj` library versioning for AIX. The **Ultra Messaging** Java API now runs on AIX.
- Corrected a race condition with the LBT-RDMA transport that could result in a seg fault if a receiver was connecting/disconnecting at the same time that the source was being deleted.
- Corrected a problem with the LBT-RDMA transport that caused a seg fault if you increased `implicit_batching_minimum_length` (`./Config/implicitbatchingoptions.html#SOURCEIMPLICITBATCHINGMINIMUMLENGTH`) from the default value.
- Corrected a problem that caused a seg fault when sending messages via the LBT-RDMA transport that are larger than approximately 3,800 bytes or less if the `transport_lbtrdma_datagram_max_size` (`./Config/transportlbt-rdmaoperationoptions.html#CONTEXTTRANSPORTLBTRDMADATAGRAMMAXSIZE`) has been reduced from the default value.
- Fixed an issue with automatic monitoring on Apple OSX that caused the process name to be set to `unknown`.



### 31.1.3. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. **29West** is investigating this issue.
- When using LBT-IPC, a seg fault can occur when sending messages larger than 65,535 bytes when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero). The seg fault occurs when fragments are lost. Setting `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced` avoids the seg fault by eliminating loss. **29West** is investigating this issue.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. **29West** is investigating this problem.
- If you use the current version of VMS (3.2.8), **LBM 4.1** issues the following warning: LOG Level 5: LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen: rdma_bind_addr failed (r=-1)`. This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.

## 31.2. UME 3.1.1

### 31.2.1. New Features

- Added a flight size mechanism that tracks messages in flight from a particular source and responds when a send would exceed the configured `ume_flight_size` (`../Config/ultramessagingpersistenceoptions.html#SOURCEUMEFLIGHTSIZE`). You can configure `ume_flight_size_behavior` (`../Config/ultramessagingpersistenceoptions.html#SOURCEUMEFLIGHTSIZEBEHAVIOR`) to either block any sends that would exceed the flight size or, allow the sends while notifying your application. For more see, **UME Flight Size** in UME Normal Operation (`../UME/ume-ops-operational-view.html#UME-OPS-NORMAL-OPERATION`).
- Added `lbm_ume_src_msg_stable()` which can mark a sequence number as stable. This may trigger a source event notification, if configured to do so and also adjusts the current number of inflight messages.

### 31.2.2. Updated Features

- Modified the `receiver-new-registration-rollback` store option's default to zero (0), which now requests no message recovery for newly registered receivers. Previously, zero (0) recovered the single latest message. Similarly, a value of 1 now recovers 1 message, whereas previously a value of 1 recovered 2 messages.

**Configuration Change Required:** Reduce this option's value by 1.

**Application Change Required:** If you use `lbm_ume_rcv_recovery_info_ex_function_cb()` in any of your applications, increase the `low_sequence_number` by one.

- Added `all-active` as a valid option for both `ume_retention_intergroup_stability_behavior` (`./Config/ultramessagingpersistenceoptions.html#SOURCEUMERETENTIONINTERGROUPSTABILITYBEHAVIOR`) and `ume_retention_intragroup_stability_behavior` (`./Config/ultramessagingpersistenceoptions.html#SOURCEUMERETENTIONINTRAGROUPSTABILITYBEHAVIOR`). An active store is defined as a registered store. A group is considered active if it has at least a quorum of active or registered stores. Intergroup stability requires at least one stable group.
- **UME** now supports a Quorum/Consensus group size of 1.
- Added two new store configuration options for separating the configuration of read async IO callbacks from and write async IO callbacks. See `repository-disk-max-write-async-cbs` and `repository-disk-max-read-async-cbs` in Options for a Topic's `ume-attributes` Element (`./UME/umestored-config.html#UMESTORED-TOPIC-OPTIONS`).
- Added the ability to track message stability and/or consumption on a per send basis as opposed to a per fragment basis by adding additional option values to `ume_message_stability_notification` (`./Config/ultramessagingpersistenceoptions.html#SOURCEUMEMESSAGESTABILITYNOTIFICATION`) and `ume_confirmed_delivery_notification` (`./Config/ultramessagingpersistenceoptions.html#SOURCEUMECONFIRMEDDELIVERYNOTIFICATION`). Values 0 (zero) and 1 retain existing behavior. Using a value of 2 provides only a single notification to the source of stability or delivery for an entire message. A value of 3 provides notification of stability or delivery for every fragment or message with the flag `WHOLE_MESSAGE_STABLE` or `WHOLE_MESSAGE_CONFIRMED` set for the last fragment of a message.
- Modified `umestored` to greatly increase the rate at which stores handle retransmission requests. Retransmissions are now sent on a different context from the store context thread.
- Changed the `IP:port` used to send stability ACKs when using **UME** across the UM Gateway. **UME** now uses the `IP:port` specified in the topic advertisement, rather than the `IP:port` specified in the persistent registration. This change only affects deployments in which
  1. The source and store are on different sides of a gateway.
  2. Multiple paths exist between the source and store via multiple parallel gateways.
  3. Due to ACL restrictions, the persistent registration and topic data are forwarded on different gateways.

### 31.2.3. Bug Fixes

- Fixed a problem that caused a segfault if `umestored` was configured to use sequential mode.
- Corrected a problem that caused proxy source creation to fail in **UME** stores.
- Fixed an issue where a persistent registration timer could be scheduled repeatedly on an error condition, which caused rapid memory growth and CPU usage.
- Fixed an issue that caused a segfault if a receiver attempted a very large recovery that completed quickly.
- Corrected an issue in the Gateway concerning multiple parallel peer portals on the same gateway which connect to corresponding peer portals on another gateway. Multiple delivery confirmations could be received for a single message due to problems with how ACLs on the peer portals operated.

- Changed the value of `optlen` to 0 (zero) if no **UME** store was configured with either a configuration file or `lbm_src_topic_attr_str_getopt()` and you called `lbm_src_topic_attr_str_getopt()`. Previously in this case, `optlen` was set to -1.
- Added a new state, **UNRESOLVED**, for stores configured with only a `store-name` and without an IP:port. Previously in this case, **UME** delivered a `LBM_SRC_EVENT_UME_STORE_UNRESPONSIVE` event to the source application. The event text now indicates that the store is **UNRESOLVED**, rather than unresponsive.
- Corrected an issue with **UME** sources referencing stores by name that have been become unresponsive and have been configured in the Round Robin fashion. Once the store restarted, re-registration did not occur. This condition only applies when the source and store reside on opposite sides of a gateway.

### 31.2.4. Known Issues

- Receivers using event queues and Spectrum with **UME** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.

## 31.3. UMQ 1.1.1

### 31.3.1. New Features

- Added a flight size mechanism for both **UMQ** and **ULB** that tracks messages in flight from a particular source and responds when a send would exceed the configured flight size. You can configure the **UMQ** flight size behavior for sources or Multicast Immediate Messaging to either block any sends that would exceed the flight size or, allow the sends while notifying your application. For more see, [UMQ Flight Size](#) (`../UME/ume-ops-operational-view.html#QUEUE-FLIGHT-SIZE`) and [Ultra Load Balancing Flight Size](#) in [Ultra Load Balancing Operations](#) (`../UME/ume-ops-operational-view.html#ULB-OPS`).
- Added `lbm_umq_ctx_msg_stable()` which can mark a message ID as stable. This may trigger a source event notification, if configured to do so and also adjusts the current number of inflight messages.

## 32. Release LBM 4.1 / UME 3.1 / UMQ 1.1 - August 2010

### 32.1. LBM 4.1

#### 32.1.1. New Features

- **New Transport.** Added the LBT-RDMA transport, which is a Remote Direct Memory Access (RDMA) **LBM** transport that allows sources to publish topic messages to a shared memory area from which receivers can read topic messages. LBT-RDMA runs across InfiniBand and 10 Gigabit Ethernet hardware. See LBT-RDMA ([../Design/lbm-objects.html#TRANSPORT-LBT-RDMA](#)) for more information.
- Added the following three new configuration options that cap the total amount of memory that a transport transmission window uses, which includes data and overhead.
  - `transport_lbtrm_transmission_window_limit` ([../Config/transportlbtrmreliabilityoptions.html#SOURCETRANSPORTLBTRMTRANSMISSIONWINDOWLIMIT](#))
  - `transport_lbtru_transmission_window_limit` ([../Config/transportlbtrureliabilityoptions.html#SOURCETRANSPORTLBTRUTRANSMISSIONWINDOWLIMIT](#))
  - `mim_transmission_window_limit` ([../Config/multicastimmediatemessagingreliabilityoptions.html#CONTEXTMIMTRANSMISSIONWINDOWLIMIT](#))

The default for all three is 0 (zero) which does not impose any limit.

- **Accelerated Multicast support.** Added another transport acceleration option to **LBM** with support for Accelerated Multicast which applies to LBT-RM. You turn this acceleration on with the `ud_acceleration` ([../Config/transportaccelerationoptions.html#CONTEXTUDACCELERATION](#)) option. This features requires InfiniBand or 10 Gigabit Ethernet hardware.

#### 32.1.2. Updated Features

- **Redesigned UM Gateway** The Gateway has been redesigned to allow full bidirectional forwarding, multi-hop forwarding, tunnel compression, encryption, and more. **This new design requires a configuration change.** For more see the new UM Gateway user guide ([../Gateway/index.html](#)).
- **Multi-Transport Threads** **LBM** has added the ability to distribute data delivery across multiple CPUs by using a receiving thread pool. Receivers created with the configuration option, `use_transport_thread` ([../Config/majoroptions.html#RECEIVERUSETRANSPORTTHREAD](#)) set to **1** use a thread from the thread pool instead of the context thread. The option, `receive_thread_pool_size` ([../Config/majoroptions.html#CONTEXTRECEIVETHREADPOOLSIZE](#)) controls the pool size. See Multi-Transport Threads ([../Design//fundamental-concepts.html#MULTITHREADED-TRANSPORTS](#)).
- Added a sequential mode for LBT-IPC which is controlled by `transport_lbtipc_receiver_operational_mode` ([../Config/transportlbtipcoperationoptions.html#CONTEXTTRANSPORTLBTIPCReceiverOPERATIONALMODE](#)). This mode requires your application to call `lbm_context_process_lbtipc_messages()` to process LBT-IPC messages instead of **LBM** automatically spawning a thread to do so.

- **New Monitoring Statistics:** Added monitoring statistics for the new LBT-RDMA transport in the `lbm_rcv_transport_stats_lbtrdma_t_stct` (`./API/structlbm_rcv_transport_stats_lbtrdma_t_stct.html`) and `lbm_src_transport_stats_lbtrdma_t_stct` (`./API/structlbm_src_transport_stats_lbtrdma_t_stct.html`) structures. These statistics are available to the C API (`./API/index.html`), the Java API (`./JavaAPI/html/index.html`) and the .NET API (`./DotNetAPI/doc/Index.html`). The new statistics have also been added to the **Ultra Messaging MIB** and the InterMapper probe files. In addition, all example applications support the statistics.
- Added the ability to name a context with `lbm_context_set_name()` and advertise it's existence at the `resolver_context_advertisement_interval` (`./Config/majoroptions.html#CONTEXTRESOLVERCONTEXTADVERTISEMENTINTERVAL`). `lbm_context_get_name()` was also added. This mechanism for naming and advertising **LBM** contexts facilitates UM Gateway operation especially for **UME** and **UMQ**.
- Zero Object Delivery (ZOD) has been implemented for .NET, which allows .NET messaging receivers to deliver messages to an application with no per-message object creation. For more information, see Zero Object Delivery (ZOD) (<https://communities.informatica.com/infakb/faq/5/Pages/80233.aspx>) in the **29West** Knowledgebase.
- Zero Incoming Copy (ZIC) has been implemented for .NET, which provides access to message data directly through a byte pointer returned by the `LBMMessage.dataPointer()` method. For more information see Zero Incoming Copy (ZIC) (<https://communities.informatica.com/infakb/faq/5/Pages/80279.aspx>) in the **29West** Knowledgebase.
- TCP-LB has now been enhanced to allow fragmented messages to be delivered as fragments when you set `ordered_delivery` (`./Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) to zero.
- Daemons built on 32-bit Linux previously linked with Smartheap 8.1 are now linked with Smartheap 9.0.1.
- Added the following five new configuration options that establish independent datagram size limits for each transport. Deprecated `transport_datagram_max_size` (`./Config/majoroptions.html#CONTEXTTRANSPORTDATAGRAMMAXSIZE`).
  - `transport_lbtrm_datagram_max_size` (`./Config/transportlbt-rmoperationoptions.html#CONTEXTTRANSPORTLBTRMDATAGRAMMAXSIZE`)
  - `transport_lbtru_datagram_max_size` (`./Config/transportlbt-ruoperationoptions.html#CONTEXTTRANSPORTLBTRUDATAGRAMMAXSIZE`)
  - `transport_tcp_datagram_max_size` (`./Config/transporttcpoperationoptions.html#CONTEXTTRANSPORTTCPDATAGRAMMAXSIZE`)
  - `transport_lbtipc_datagram_max_size` (`./Config/transportlbt-ipcoperationoptions.html#CONTEXTTRANSPORTLBTIPCDATAGRAMMAXSIZE`)
  - `transport_lbtrdma_datagram_max_size` (`./Config/transportlbt-rdmaoperationoptions.html#CONTEXTTRANSPORTLBTRDMADATAGRAMMAXSIZE`)

The minimum size is 500 bytes for all datagram size options including `resolver_datagram_max_size` (`./Config/resolveroperationoptions.html#CONTEXTRESOLVERDATAGRAMMAXSIZE`).
- Added the ability to continually report statistics based on a saved search term, such as `transport ID` to the example application, **lbmoncache.c**. Also updated the example applications, **lbmoncache.c** and **lbmon.c** to accept the `-c config` argument. Also added `lbm_context_topic_resolution_request()` to **lbmoncache.c** to help resolve any quiescent topics.

- Updated event processing to prevent short timers, or a large number of timers expiring at the same time from starving network processing. Network processing is now interspersed with timer expirations under such conditions.

### 32.1.3. Bug Fixes

- Corrected an incompatibility problem between **LBM 4.0** and pre-4.0 versions. When running **LBM 4.0**, any pre4.0 receivers were unable to discover multiple sources of the same topic. 4.0 receivers were able to discover all sources sending on the same topic. **LBM 4.1** resolves this problem so all receivers discover all sources sending on the same topic.
- Corrected a condition where a particular sequence of `lbm_src_topic_alloc`, `lbm_src_delete`, and `lbm_src_create` API calls would result in a fatal assertion.
- Corrected a condition that caused a seg fault when using LBT-IPC to send large, fragmented messages (approximately over 65,000 bytes) and `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDELIVERY`) was set to 0 (zero).
- Corrected a problem that caused spurious context source wakeup events to be delivered for Unicast Immediate Messaging (UIM) when the immediate messaging had never actually been blocked.
- Corrected a problem that sometimes resulted in a seg fault if you delete a source immediately after sending a message. The seg fault occurred when **LBM** flushed the messages for that source out of the batch.
- Changed `lbm_context_topic_resolution_request()` to implement 0 (zero) for the `duration_sec` parameter which results in only one request sent.
- Changed the handling of arrival order and arrival order reassembly to expire records about lost packets under some conditions. This change rectifies cases where the last set of lost packets on a stream would not be reported as unrecoverably lost before the loss disappeared. With arrival order reassembly, unrecoverable loss was rarely reported.
- Fixed some issues with Java context statistics methods that could result in invalid memory being accessed.
- Fixed an issue where calling the `LBMContext.getReceiverStatistics()` could cause an access violation.

### 32.1.4. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. **29West** is investigating this issue.
- When using LBT-IPC, a seg fault can occur when sending messages larger than 65,535 bytes when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDELIVERY`) has been set to 0 (zero). The seg fault occurs when fragments are lost. Setting `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced` avoids the seg fault by eliminating loss. **29West** is investigating this issue.
- When using the LBT-RDMA transport with Java applications, a segfault can occur if you kill a receiver with **Ctrl-C**. As a workaround, use the JVM option, `-Xrs`. **29West** is investigating this problem.
- If you use the current version of VMS (3.2.8), **LBM 4.1** issues the following warning: LOG Level 5:  
LBT-RDMA: VMS Logger Message (Error): `vmss_create_store: 196[E] vms_listen:`

`rdma_bind_addr` failed (`r=-1`). This warning indicates that `rdma_bind` failed for ethernet interfaces, which is expected behavior. Currently, VMS attempts `rdma_bind` on all interfaces. When released, VMS version 3.2.9 will only run `rdma_bind` on infiniband-capable interfaces.

## 32.2. UME 3.1

### 32.2.1. Updated Features

- Updated the `umestored` `<interface>` attribute of a `<store>` or `<queue>` element to use CIDR notation, i.e., 10.29.3.0/24.
- Both **UME** and **UMQ** now deliver the `Registration Complete` source event every time a quorum is established instead of only once. More than one event delivery indicates quorum was lost and re-established.
- Implemented an internal function to offload all non-critical log messages to a separate thread to avoid blocking I/O on critical threads inside the `umestored` and `umestoreds` daemons.

### 32.2.2. Bug Fixes

- Fixed a problem that caused a **UME** source to erroneously register with one more configured store than needed after a failover when using quorum/consensus and the number of stores configured in the group is greater than the configured group size.

### 32.2.3. Known Issues

- Receivers using event queues and Spectrum with **UME** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.

## 32.3. UMQ 1.1

### 32.3.1. New Features

- Added a new feature called Ultra Load Balancing (ULB) that provides Once And Only Once (OAOO) delivery to receiving applications, but without a queue. The lack of a queue or broker in the message path adds ultra low latency to the load balancing abilities of this **UMQ** feature. Sources perform message assignment, delivery is

receiver-paced and messages are not persisted. For more, see The Ultra Messaging, Queuing Edition User Guide ([../UME/index.html](#)).

- **UMQ** now supports messages with user-supplied, chained application headers.
- Added support for user-defined message headers, called **application headers**, to the C and Java APIs. Application headers are optional and reside outside the normal message payload.

### 32.3.2. Bug Fixes

- Corrected a condition that caused the Unicast Topic Resolver (`lbmrtd`) to segfault when processing **UMQ** store advertisements.

## 33. Release LBM 4.0.1 - June 2010

### 33.1. LBM 4.0.1

#### 33.1.1. Bug Fixes

.

## 34. Release LBM 3.6.6 / UME 3.0.1 - August 2010

### 34.1. LBM 3.6.6

#### 34.1.1. Bug Fixes

- Updated datagram transports to allocate buffers based on `transport_datagram_max_size` ([../Config/majoroptions.html#CONTEXTTRANSPORTDATAGRAMMAXSIZE](#)) instead of the maximum size of 65535 bytes. This reduces the memory footprint of processes when `transport_datagram_max_size` ([../Config/majoroptions.html#CONTEXTTRANSPORTDATAGRAMMAXSIZE](#)) is set less than the maximum (65535), such as the default of 8192. Note that sources and receivers that communicate in a system should now be configured with identical `transport_datagram_max_size` ([../Config/majoroptions.html#CONTEXTTRANSPORTDATAGRAMMAXSIZE](#)) settings.



## 35. Release LBM 3.6.5 - May 2010

### 35.1. LBM 3.6.5

#### 35.1.1. Bug Fixes

- Fixed an issue that caused Adaptive Batching to always send messages immediately during periods of high volume, instead of the expected behavior of batching messages during high volume.
- Fixed an issue that caused unicast immediate messages sent from a Java application to return a fatal assert.

## 36. Release LBM 4.0 - May 2010

### 36.1. LBM 4.0

#### 36.1.1. New Features

- **LBM** now supports Datagram Bypass Layer (DBL) transport acceleration in conjunction with DBL-enabled Myricom® (<http://www.myri.com>) 10-Gigabit Ethernet NICs for Linux and Microsoft Windows. DBL is a kernel-bypass technology that accelerates sending and receiving UDP traffic. See Transport Acceleration Options ([../Config/transportaccelerationoptions.html](http://Config/transportaccelerationoptions.html)) for more information. **Note:** The initial valid version of the Myricom DBL shared library is Version 0.4.8. Do not download any version prior to this version.

#### 36.1.2. Updated Features

- **29West** topic resolution has been enhanced to provide more flexibility. Three phases have been implemented.
  - Initial Phase - Period that allows you to resolve a topic aggressively. Can be used to resolve all known topics before message sending begins. This phase can be configured to run differently from the defaults or completely disabled.
  - Sustaining Phase - Period that allows new receivers to resolve a topic after the Initial Phase. This phase is most like the existing topic resolution and can also be the primary period of topic resolution if you disable the Initial Phase. It can also be configured to run differently from the defaults or completely disabled.
  - Quiescent Phase - The "steady state" period during which a topic is resolved and **LBM** uses no system resources for topic resolution.

See Topic Resolution ([../Design/architecture.html#TOPIC-RESOLUTION](http://Design/architecture.html#TOPIC-RESOLUTION)) for more information.

- Two changes have been made to the LBT-IPC transport.

- Receiver pacing has been added to the LBT-IPC transport. When the IPC shared memory area is full, sources either block or return an error (EWOULDBLOCK). You configure source or receiver pacing with `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#RECEIVERTRANSPORTLBTIPCBEHAVIOR`).
- **LBM** now monitors receiver "health". If a receiver goes away, the source receives a disconnect event the next time the sending application sends a message. As a result, the LBT-IPC Shared Memory Layout now has a lock-free design and the configuration options, `transport_lbtipc_client_activity_timeout` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCCLIENTACTIVITYTIMEOUT`) and `transport_lbtipc_acknowledgement_interval` (`../Config/transportlbt-ipcoperationoptions.html#RECEIVERTRANSPORTLBTIPCACKNOWLEDGEMENTINTERVAL`) have been deprecated.

See Transport LBT-IPC (`../Design/lbm-objects.html#TRANSPORT-LBT-IPC`) for more information.

- In the **C API**, the previously deprecated functions have been removed.

```
lbm_context_attr_init()
lbm_event_queue_attr_init()
lbm_rcv_topic_attr_init()
lbm_src_topic_attr_init()
lbm_wildcard_rcv_attr_init()
lbm_context_attr_cleanup()
lbm_event_queue_attr_cleanup()
lbm_rcv_topic_attr_cleanup()
lbm_src_topic_attr_cleanup()
lbm_wildcard_topic_attr_cleanup()
```

- In the **Java API**, the previously deprecated JAVA APIs have been removed.

LBMSource:

```
LBMSource(LBMContext lbmctx, LBMTopic lbmtopic, LBMSourceCallback cb, Object cbArg)
LBMSource(LBMContext lbmctx, LBMTopic lbmtopic, LBMSourceCallback cb, Object cbArg,
    LBMEventQueue lbmevq)
```

```
addSourceCallback(LBMSourceCallback cb)
addSourceCallback(LBMSourceCallback cb, Object cbArg)
removeSourceCallback(LBMSourceCallback cb)
removeSourceCallback(LBMSourceCallback cb, Object cbArg)
onSourceEvent(int event, byte [] eventData)
send(byte [] data, int dataLength, LBMSourceCallback cb, LBMEventQueue lbmevq, int flags)
send(byte [] data, int dataLength, LBMSourceCallback cb, int flags)
```

LBMContext:

```
setCallback(LBMImmediateMessageCallback cb, Object cbArg)
setCallback(LBMImmediateMessageCallback cb, Object cbArg, LBMEventQueue lbmevq)
createSource(LBMTopic lbmtopic, LBMSourceCallback cb, Object cbArg)
createSource(LBMTopic lbmtopic, LBMSourceCallback cb, Object cbArg, LBMEventQueue lbmevq)
```

LBMContextAttributes:

```
setSourceNotifyCallback(LBMSourceNotification cb, Object cbArg)
```

LBMEventQueue:

```
size()
```

LBMSourceCallback: (entire interface)

**The previously deprecated JAVA APIs have been undeprecated.**

```
LBMEventQueue:
LBMEventQueue(LBMEventQueueCallback cb, Object cbArg)
LBMEventQueue(LBMEventQueueAttributes lbmevqattr, LBMEventQueueCallback cb, Object cbArg)
LBMContext:
send(String target, String topic, byte [] data, int dataLength, LBMSourceCallback cb,
    Object cbArg, LBMEventQueue lbmevq, int flags)
send(String target, String topic, byte [] data, int dataLength, LBMSourceCallback cb,
    Object cbArg, int flags)
send(String topic, byte [] data, int dataLength, LBMSourceCallback cb, Object cbArg,
    LBMEventQueue lbmevq, int flags)
send(String topic, byte [] data, int dataLength, LBMSourceCallback cb, Object cbArg, int flags)
createReceiver(LBMTopic lbmtopic, LBMSourceCallback cb, Object cbArg)
createReceiver(LBMTopic lbmtopic, LBMSourceCallback cb, Object cbArg, LBMEventQueue lbmevq)
```

**The following JAVA APIs have been deprecated. A null pointer check has also been added to**

```
LBMEventQueue.addMonitor().
LBMEventQueue.propertySize()
LBMContext:
createReceiver(LBMTopic lbmtopic)
createReceiver(LBMTopic lbmtopic, LBMEventQueue lbmevq)
createHotFailoverReceiver(LBMTopic lbmtopic, LBMEventQueue lbmevq)
createHotFailoverReceiver(LBMTopic lbmtopic)
```

**The following JAVA APIs have been changed from public to protected.**

```
LBMSource(LBMContext lbmctx, LBMTopic lbmtopic)
LBMSource(LBMContext lbmctx, LBMTopic lbmtopic, LBMEventQueue lbmevq)
LBMSourceBase(LBMContext lbmctx, LBMSourceAttributes lbmsrcvattr, LBMEventQueue lbmevq)
LBMSourceBase(LBMContext lbmctx, LBMSourceAttributes lbmsrcvattr)
LBMSourceHotFailoverReceiver(LBMContext lbmctx, LBMTopic lbmtopic)
LBMSourceHotFailoverReceiver(LBMContext lbmctx, LBMTopic lbmtopic, LBMEventQueue lbmevq)
```

- **In the .NET API, the previously deprecated .NET APIs have been removed.**

```
LBMSource:
addSourceCallback(LBMSourceCallback cb)
addSourceCallback(LBMSourceCallback cb, object cbArg)
removeSourceCallback(LBMSourceCallback cb)
removeSourceCallback(LBMSourceCallback cb, object cbArg)
LBMSource(LBMContext lbmctx, LBMTopic lbmtopic, LBMSourceCallback cb, object cbArg)
LBMSource(LBMContext lbmctx, LBMTopic lbmtopic, LBMSourceCallback cb, object cbArg,
    LBMEventQueue lbmevq)
delegate void LBMSourceCallback(object cbArg, int evtype, byte[] eventData)

LBMContext:
createSource(LBMTopic lbmtopic, LBMSourceCallback cb, object cbArg)
createSource(LBMTopic lbmtopic, LBMSourceCallback cb, object cbArg, LBMEventQueue lbmevq)
```

```
LBMContextAttributes:
setSourceNotifyCallback(LBMSourceNotification cb, object cbArg)
```

The following .NET APIs have been deprecated.

```
LBMContext:
createReceiver(LBMTopic lbmtopic)
createReceiver(LBMTopic lbmtopic, LBMEventQueue lbmevq)
createHotFailoverReceiver(LBMTopic lbmtopic)
createHotFailoverReceiver(LBMTopic lbmtopic, LBMEventQueue lbmevq)
```

The following .NET APIs have been changed from public to protected.

```
LBMReceiver(LBMContext lbmctx, LBMTopic lbmtopic)
LBMReceiver(LBMContext lbmctx, LBMTopic lbmtopic, LBMEventQueue lbmevq)
LBMHotFailoverReceiver(LBMContext lbmctx, LBMTopic lbmtopic)
LBMHotFailoverReceiver(LBMContext lbmctx, LBMTopic lbmtopic, LBMEventQueue lbmevq)
```

- Added `const` qualifier to `optval` parameters of all `lbm*_attr_setopt()` API functions. Added `const` qualifier to attributes pointer parameters of functions used to create **LBM** objects such as `lbm_context_create()`, `lbm_src_create()`, `lbm_rcv_create()`, etc.
- **Configuration Change Required for LBM 3.5.3 / UME 2.2.4 or earlier:** Added the Configuration Option, `disable_extended_topic_resolution_message_options` (`./Config/resolveroperationoptions.html#CONTEXTDISABLEEXTENDEDTOPICRESOLUTIONMESSAGEOPTIONS`), for backward compatibility. If you use **LBM 3.5.3 / UME 2.2.4** or less, you should set this option to "1" to disable the internal topic resolution message options. Not disabling these message options produces warnings and deaf receivers. No action is required for **LBM 3.6 / UME 3.0** and later.
- Added the Configuration Options, `mim_delivery_control_activity_check_interval` (`./Config/multicastimmediatemessagingoperationoptions.html#CONTEXTMIMDELIVERYCONTROLACTIVITYCHECKINTERVAL`) and `mim_delivery_control_activity_timeout` (`./Config/multicastimmediatemessagingoperationoptions.html#CONTEXTMIMDELIVERYCONTROLACTIVITYTIMEOUT`) to check for duplicate MIM messages forwarded through multiple Gateways.
- Added a new statistic in `lbm_rcv_transport_stats_lbtrm_t` (`./API/structlbm_rcv_transport_stats_lbtrm_t_stct.html`) structure for datagrams received out of order. The new statistic is available to the C API (`./API/index.html`), the Java API (`./JavaAPI/html/index.html`) and the .NET API (`./DotNetAPI/doc/Index.html`). The statistic has also been added to the **Ultra Messaging MIB** and the InterMapper probe files. In addition, all example applications support the statistic.
- Blocking sends from within a context-thread callback are no longer permitted and now return an `LBM_EINVAL` error. This restriction eliminates a possible deadlock. Non-blocking sends from a context-thread callback are still permitted, and any type of send is permitted outside a context-thread callback.
- Some API calls that were formerly never safe to call from within a context-thread callback (due to a deadlock condition) are now always safe to call from within a context-thread callback. These API calls include:
  - `lbm_request_delete()`
  - `lbm_request_delete_ex()`
  - `lbm_response_delete()`

Additionally, the following send-related API calls are now safe to call from within a context-thread callback provided you set the `LBM_SRC_NONBLOCK` flag.

- `lbm_multicast_immediate_message()`
  - `lbm_multicast_immediate_request()`
  - `lbm_unicast_immediate_message()`
  - `lbm_unicast_immediate_request()`
  - `lbm_send_response()`
- Added ability to set a **LBM** Configuration Option interface by name. When setting by name the configuration option's value must be wrapped in quotes.

### 36.1.3. Bug Fixes

- Corrected the installation of 64 bit **LBM** packages. **LBM** now installs into `C:\Program Files` instead of `C:\Program Files (x86)`.
- All include files now reference other include files using `lbm/...`. This allows your application Makefiles to specify just the parent directory, e.g. `/usr/include`. You can copy the `include/lbm` directory as a whole to your common or system include directory.
- Added a warning "Data received but no callback registered" when data is received and no callbacks are registered in the Java and .NET APIs.
- Fixed a bug that occurred when using Microsoft Windows IO Completion Ports that could cause a context's `request_tcp_port` to become permanently unresponsive to incoming TCP connection requests. This condition manifests itself in many ways, such as an inability to participate in **LBM** Request/Response, Late Join, or an inability to receive Unicast Immediate Messages. This condition occurs if the listening TCP socket created by **LBM** (e.g. by multiple incoming responses or UIMs) receives multiple TCP connection requests. See also *UME* 2.2.3.
- Changed type of `queue_size_warning` (`./Config/eventqueueoptions.html#EVENTQUEUEQUEUESIZEWARNING`) from `unsigned long` to `size_t`.
- Fixed an issue where uninitialized fields in an `lbm_msg_t` passed a bad pointer to application code. This could sometimes result in exceptions in .NET and Java. The fields in question were only uninitialized in unrecoverable loss messages larger than the `transport_datagram_max_size` (`./Config/eventqueueoptions.html#EVENTQUEUEQUEUESIZEWARNING`) and delivered from hot failover receivers or immediate messages. This issue was introduced in **LBM** 3.6.
- Added the option, `resolver_unicast_force_alive` (`./Config/resolveroperationoptions.html#CONTEXTRESOLVERUNICASTFORCEALIVE`), which allows an **LBM** Gateway to begin sending keepalive messages to a Unicast Topic Resolution daemon (`lbmrd`). The `lbmrd` can then begin sending topic resolution traffic to the **LBM** Gateway.
- Fixed an issue that caused unicast immediate messages sent from a Java application to return a fatal assert.

### 36.1.4. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. **29West** is investigating this issue.
- When using LBT-IPC, a seg fault can occur when sending messages larger than 65,535 bytes when `ordered_delivery` (`../Config/majoroptions.html#RECEIVERORDEREDEDELIVERY`) has been set to 0 (zero). The seg fault occurs when fragments are lost. Setting `transport_lbtipc_behavior` (`../Config/transportlbt-ipcoperationoptions.html#SOURCETRANSPORTLBTIPCBEHAVIOR`) to `receiver_paced` avoids the seg fault by eliminating loss. **29West** is investigating this issue.

## 37. Release LBM 3.6.3 / UME 3.0.1 / UMQ 1.0 - April 2010

### 37.1. LBM 3.6.3

#### 37.1.1. Bug Fixes

- Fixed an issue that resulted in **LBM** sending additional topic queries and delivering erroneous `NO_SOURCE_NOTIFICATION` if an application created multiple receivers on a single topic in a short period of time.

### 37.2. UME 3.0.1

#### 37.2.1. Bug Fixes

- Corrected the installation of 64 bit **UME** packages. **UME** now installs into `C:\Program Files` instead of `C:\Program Files (x86)`.
- Fixed an issue that prevented the creation of proxy sources when using multiple store groups. This issue was indicated by the error message, `[ERROR] store "<store name>" topic "<topic name>" Proxy creation failed to set ume_store_group, optlen incorrect size.`

#### 37.2.2. Known Issues

- Receivers using event queues and Spectrum with **UME** can experience a `SIGSEGV` while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the

event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.

## 37.3. UMQ 1.0

### 37.3.1. Known Issues

Release 1.0 of UMQ has no known issues.

## 38. Release LBM 3.6.2 / UME 3.0 - February 2010

### 38.1. LBM 3.6.2

#### 38.1.1. Bug Fixes

- Fixed a problem in Java and .NET that produced an exception when uninitialized fields in an `lbm_msg_t` caused a bad pointer to be passed to your application code. The fields in question were only uninitialized in messages of unrecoverable loss that were delivered from hot failover receivers or immediate messages larger than the `transport_datagram_max_size` (`../Config/config.html#CONTEXTTRANSPORTDATAGRAMMAXSIZE`) setting. This issue was introduced in LBM 3.6.

### 38.2. UME 3.0

See also *LBM 3.6.1*.

#### 38.2.1. New Features

- Added the **Proxy Source** feature that allows you to configure stores to automatically assume sending activity for a source or sources that cease operation. After the source returns, the store automatically stops acting as a proxy source. The main benefit of a Proxy Source is the continuation of source's topic advertisements which contain store information used by new receivers. Without the store Reg ID, address and TCP port contained in the source's Topic Information Records (TIR), new receivers cannot register with the store or request retransmissions. Proxy Source requires a Quorum/Consensus store configuration. See Proxy Sources (`../UME/ume-fault-tolerance.html#PROXYSOURCES`) for more information.
- Added large file support to the UME store on 32-bit and 64-bit Windows, and 64-bit Linux and Solaris platforms. The maximum size of individual disk cache files (controlled by the "repository-disk-file-size-limit" store configuration option) can now exceed 4GB.

- To reduce disk fragmentation, the UME store can now optionally pre-allocate disk cache files at source registration time. To enable disk file pre-allocation, set the new `repository-disk-file-preallocate` store configuration option. See Configuration Reference for Umestored ([../UME/umestored-config.html](#)) for more information.
- Added the ability to batch stability acknowledgments sent to sources from the UME store. In some cases -- especially with a memory store -- this feature can increase overall throughput, at the cost of a small delay between the time a message is actually stable at the store and the time the source is informed of message stability. Stability ACK batching behavior can be controlled via two new UME store topic configuration options, `stability-ack-minimum-number` and `stability-ack-interval`. Please see Configuration Reference for Umestored ([../UME/umestored-config.html](#)) for a description of the new options.

### 38.2.2. Bug Fixes

- Improved the ability of a Store in a Quorum/Consensus group to handle large amounts of loss. Previously, if a Store suspended operation and a source continued sending while the Store was down, the Store would perpetually handle loss after resuming operation with a high CPU consumption. The Store now processes loss messages more efficiently.
- Fixed a bug that could cause a UME store to consume a lot of CPU at startup during a period of initially heavy loss.
- Fixed a bug that could cause a UME store to crash at startup if using a configuration file containing unknown options.
- Fixed handling of "connection timedout" socket errors for Linux to Microsoft Windows TCP connections. These errors occurred on a Linux `umestored` process after a Microsoft Windows **UME** source or receiver was terminated.

### 38.2.3. Known Issues

- Receivers using event queues and Spectrum with **UME** can experience a SIGSEGV while shutting down if events still exist on the event queue when it is deleted. As a workaround, use `LBM_EVQ_BLOCK` when dispatching event queues. During application shutdown, call `lbm_evq_unblock()` after deleting receivers associated with the event queue, but before deleting any context objects. Once the dispatch thread exits, it is safe to proceed with context deletion. 29West is working on a solution to this problem.

## 39. Release LBM 3.6.1 - February 2010

### 39.1. LBM 3.6.1

#### 39.1.1. New Features

- Changed the naming of `lbmmon` statistics to refer to datagrams rather than messages. For example, the LBT-RM



transport receiver statistic for **bytes received** is now labeled, **datagram bytes received** because the statistic includes datagram overhead bytes in addition to application message bytes.

- Implemented backward compatibility for LBMMON statistics packets for **LBM** Version 3.6.1 and later by enhancing LBMMON internal message processing. Backward statistics compatibility does not apply to prior **LBM** versions.
- Added support for multiple outstanding asynchronous AcceptEx calls when using Microsoft Windows IO completion ports. This allows multiple incoming connections to be serviced more quickly when using completion ports. The number of outstanding AcceptEx calls can be controlled by the `transport_tcp_listen_backlog` (`./Config/transporttcpoperationoptions.html#SOURCETRANSPORTTCPLISTENBACKLOG`) and `request_tcp_listen_backlog` (`./Config/requestoperationoptions.html#CONTEXTREQUESTTCPLISTENBACKLOG`) configuration options.
- **LBM** now logs the originating IP and port number of all non-LBM packets received.
- Changed the previously internal message `timestamp` field to a public field in the `lbm_msg_t` (`./API/structlbn__msg__t__stct.html`) structure. Also added the `timestampSeconds()` and `timestampMicroseconds()` methods to .NET and Java `LBMMessage` classes.
- Added the following methods to the Java API and the .NET API.
  - `LBMContext sendc(char [] target, char [] topicname, byte [] data, int datalen, int flags)` for sending unicast immediate messages where both `target` and `topicname` are character arrays.
  - `LBMContext sendc(char [] topicname, byte data, int datalen, int flags)` for sending multicast immediate messages where the `topicname` is a character array.
  - `LBMContext sendTopicless(byte [] data, int datalen, int flags)` for sending topicless multicast immediate messages.
  - `LBMContext sendTopicless(String target, byte [] data, int datalen, int flags)` for sending topicless unicast messages using a string as the target.
  - `LBMContext sendTopicless(char [] target, byte [] data, int datalen, int flags)` for sending topicless unicast messages using a `char []` as the target.
  - `LBMMessage char [] sourceAsCharArray()` for returning the source of a message as a character array.
  - `LBMMessage char [] topicNameAsCharArray()` for returning the `topicname` of a message as a character array.
  - `LBMMessage char [] originalSourceAsCharArray()` for returning the original source (if message was from LBM Gateway) as a character array.

### 39.1.2. Bug Fixes

- Corrected a problem with fragmented MIM messages sent to a topic and received by a receiver configured for Arrival Order (`mim_ordered_delivery` (`./Config/multicastimmediatemessagingoperationoptions.html#CONTEXTMIMORDEREDELIVERY`) = 0). The MIM message fragments were not properly delivered to the topic receiver.

- **Clarification:** In the `toString` method for the Java and .Net `LBMSDMRawTimestamp` object, the value returned by `LBMSDMRawTimestamp` is not intended to be a floating point representation of the time. The value is simply formatted as `seconds.microseconds`.
- Added support on HP-UX for the EAGAIN error during message sends by treating it as an EWOULDBLOCK.
- Corrected a problem with the LBM Gateway that caused the LBM Gateway to assert after receiving a `topic leave` message from a receiver shutting down normally. Now the LBM Gateway does not assert or deadlock on normal receiver shutdowns.
- Corrected a problem with how the length passed to `lbm_transport_source_parse()` and `lbm_transport_source_format()` was interpreted. The incorrect interpretation caused these functions to return an error.
- Corrected an output labeling problem with `lbmmon.cs` and `lbmmon.java`. LBT-RM receiver datagram statistics were displayed as LBT-RU statistics.
- Corrected errors in the source string of the `lbm_src_transport_stats_lbtru_t` (`./API/structlbm_src_transport_stats_lbtru_t.stct.html`) structure. Statistics were reported correctly, but the source was not properly identified. This error has also been observed and corrected for Immediate Messaging source statistics.
- Corrected a problem with the LBM MON API that resulted in several fields in the LBT-RM and LBT-RU receiver statistics to be handled improperly.
- The `lbmaux` library APIs (`./API/lbmaux_8h.html`) for loading attributes directly from **29West** Configuration Files was updated to use common code with the **LBM** library. This resolves a problem where some configuration options were not parsed correctly in `lbmaux` library APIs (`./API/lbmaux_8h.html`) but parsed correctly in **LBM**.
- Fixed a potential deadlock when context creation fails for an embedded mode context using Microsoft Windows Completion Ports.

### 39.1.3. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. **29West** is investigating this issue.
- Shutting down a source-side LBM Gateway with Control-C can cause the Gateway to become dead locked. **29West** is investigating this issue.
- If a source LBM Gateway was shutdown and then the receiving Gateway was shutdown as quickly as possible while still carrying traffic, a fatal assert occurs. (`MUL_FATAL_ASSERT (Table!=NULL) ;`) **29West** is investigating this issue.
- With **LBM 3.6** we have seen increased memory utilization in varying degrees across all platforms. This increase can be mitigated by the use of SmartHeap. **29West** is working to decrease this memory utilization.
- When turning on Automatic Monitoring, be sure that your application does not have any existing `lbmmon` monitoring capability active. Using both Automatic Monitoring and `lbmmon` functions results in confusion with internal monitoring fields and pointers. This issue will be addressed in a future release.

## 40. Release LBM 3.6 - December 2009

### 40.1. LBM 3.6

#### 40.1.1. New Features

- Implemented arrival order reassembly when message fragments are received out of order.
- Added support for wildcard receivers to the LBM Gateway. See the LBM Gateway documentation for more information.
- Added the source configuration option, `retransmit_retention_age_threshold` (`../Config/latejoinoptions.html#SOURCERETRANSMITRETENTIONAGETHRESHOLD`), to allow age-based message reclamation from a source's late join retention buffer.
- Modified the configuration option, `transport_lbtrm_destination_port` (`../Config/transportlbt-rmnetworkoptions.html#SOURCETRANSPORTLBTRMDESTINATIONPORT`), to also determine when **LBM** should create a unique LBT-RM transport session. Previously, only the LBT-RM Multicast Address as defined by `transport_lbtrm_multicast_address` (`../Config/transportlbt-rmnetworkoptions.html#SOURCETRANSPORTLBTRMMULTICASTADDRESS`) (source), `transport_lbtrm_multicast_address_high` (`../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMMULTICASTADDRESSHIGH`) (context) and/or `transport_lbtrm_multicast_address_low` (`../Config/transportlbt-rmnetworkoptions.html#CONTEXTTRANSPORTLBTRMMULTICASTADDRESSLOW`) (context) was used to determine unique transport sessions.
- Increased the maximum value for `transport_datagram_max_size` (`../Config/majoroptions.html#CONTEXTTRANSPORTDATAGRAMMAXSIZE`) to 65535 bytes. This enables large messages to be fragmented into larger fragments reducing the number of system calls per message.
- Added a LBM Gateway log message that indicates if a user requested a Gateway shutdown (`[timestamp] Shutdown event detected. Attempting to shutdown.`).
- Added version and package information to the Java **LBM** jar file. Also added an initialization-time warning if the Java **LBM** jar file and `lbmj` native library versions are not compatible.
- Added the configuration option, `resolver_datagram_max_size` (`../Config/resolveroperationoptions.html#CONTEXTRESOLVERDATAGRAMMAXSIZE`) to control the maximum datagram size that can be generated for topic resolution advertisements and queries.
- Enabled the `callback_events` statistics (See *LBM 3.6eal New Features*) to be understood by a LBMMON monitor receiver. Also enabled these statistics in all LBMMON example applications, the **Ultra Messaging MIB** and the InterMapper probe files.
- Modified the **LBM** Topic Structure, returned from `lbm_rcv_topic_lookup()` or `lbm_src_topic_allocate()`, so that it uses less memory.
- Replaced certain locks on Solaris Sparc with atomic instructions to increase performance.
- Added the ability to change a wildcard receiver's attributes using `lbm_wildcard_rcv_attr_setopt()`. The **29West** Configuration Option, `resolver_query_max_interval`

(../Config/wildcardreceiveroptions.html#WILDCARDRECEIVERRESOLVERQUERYMAXINTERVAL) can now be set during operation.

- Increased the maximum number of LBM message fragments that can be sent in a single socket send call from 16 to 1024 for Solaris and AIX. This should provide a significant throughput increase for small message sizes when batching is enabled.
- Added the source configuration option, `transport_tcp_coalesce_threshold` (../Config/transporttcpoperationoptions.html#SOURCETRANSPORTTCP\_COALESCE\_THRESHOLD) that configures the maximum number of **LBM** message fragments sent in a single socket send call without coalescing.
- Added the environment variable, `LBM_CONTEXT_DELAY_WARNING_THRESHOLD`, which specifies in milliseconds a context thread delay threshold. If a context thread is delayed for a time greater than the threshold, **LBM** logs a warning message.
- Corrected a race condition that could occur if a source was sending while another source was joining or leaving the same transport session.

### 40.1.2. Bug Fixes

- Fixed a problem that occurred when using `LBM_MSG_COMPLETE_BATCH` or `LBM_MSG_END_BATCH` flag with `lbm_src_sendv()`. This combination resulted in an error return (-1) with the error string "Explicit Batch not started".
- Corrected a problem with high resolution timers used to perform microsecond measurements which could cause some Event Queue statistics, such as the `age_max` statistic to report huge values. This correction results in a zero being returned instead of a negative value in the event that an ending timestamp is earlier than a beginning timestamp.
- Corrected a problem with using `TCP_NODELAY` that caused a request response to fail. **LBM** now sets `TCP_NODELAY` before the TCP connection call.
- Corrected a problem with Late Join that caused a delay if a source no longer had a requested message available in its Late Join retention buffer. The delay no longer occurs.
- Corrected a problem with **LBM** Spectrum wildcard receivers that caused them to crash if they were created with a NULL topic attributes object and `unrecognized_channel_behavior` (../Config/deliverycontroloptions.html#RECEIVERUNRECOGNIZEDCHANNELBEHAVIOR) or `null_channel_behavior` (../Config/deliverycontroloptions.html#RECEIVERNULLCHANNELBEHAVIOR) were set to `discard` in a **29West** Configuration File. This was specified as a *LBM 3.6ea2 Known Issues*.
- Corrected a problem with wildcard receivers in Java that prevented them from being garbage collected. Global references for the internal `onReceiverDelete` callback are now released, allowing garbage collection of the Java wildcard receivers.
- Corrected a problem with the `LBMSDMArray*` classes and the `LBMSDMFieldIterator` class in both Java and .NET.
- Fixed an incorrect pointer reference in the internal Java function that creates a receiver. This was required because topic attributes have been changed to pointers in **LBM 3.6**.
- Fixed a memory leak that could occur when creating a receiver using the Java API without setting per-source callbacks.

- Corrected a second memory leak that could occur when creating a receiver using the Java API without specifying receiver attributes.
- Modified the configuration options `transport_lbtrm_coalesce_threshold` (`./Config/transportlbt-rmoperationoptions.html#SOURCETRANSPORTLBTRMCOALESCETHRESHOLD`) and `transport_lbtru_coalesce_threshold` (`./Config/transportlbt-ruoperationoptions.html#SOURCETRANSPORTLBTRUCOALESCETHRESHOLD`) so their maximum values correspond to platform-specific maximums.
- Adjusted handling of LBT-RM NAK backoff timers to improve performance in heavy loss scenarios.

### 40.1.3. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. **29West** is investigating this issue.
- Shutting down a source-side LBM Gateway with Control-C can cause the Gateway to become dead locked. **29West** is investigating this issue.
- If a source LBM Gateway was shutdown and then the receiving Gateway was shutdown as quickly as possible while still carrying traffic, a fatal assert occurs. (`MUL_FATAL_ASSERT (Table != NULL) ;`) **29West** is investigating this issue.
- With **LBM 3.6** we have seen increased memory utilization in varying degrees across all platforms. This increase can be mitigated by the use of SmartHeap. **29West** is working to decrease this memory utilization.
- When turning on Automatic Monitoring, be sure that your application does not have any existing `lbmmon` monitoring capability active. Using both Automatic Monitoring and `lbmmon` functions results in confusion with internal monitoring fields and pointers. This issue will be addressed in a future release.

## 41. Release UME 2.2.4 - November 2009

### 41.1. Bug Fix

- Fixed a very rare bug that could cause a receiver to register with a **UME** store using an incorrect Reg ID if registration information changed in the middle of the receiver's registration attempt.

## 42. Release LBM 3.5.3 / UME 2.2.3 - November 2009

### 42.1. LBM 3.5.3

#### 42.1.1. Bug Fixes

- Corrected a problem that prevented a requestor from receiving responses from a Multicast Immediate Message request if the `request_tcp_interface` (`./Config/requestnetworkoptions.html#CONTEXTREQUESTTCPINTERFACE`) differs from the `resolver_multicast_address` (`./Config/multicastresolvernetworkoptions.html#CONTEXTRESOLVERMULTICASTADDRESS`).
- Fixed a problem in .NET when using SDM Unicode strings that produced an error similar to, `Failed: sdm_unicode_src1 Not enough space in buffer to format (offset 28 length 30 needed 5)`. The `format()` method has been changed to encode the Unicode string length before it performs the storage length check.

### 42.2. UME 2.2.3

#### 42.2.1. Bug Fixes

- Corrected a problem when recovering old messages from a source or UME Store using Late Join or UME that caused the recovery process to either stop delivering messages to the application or delay delivery for a long period of time. This disruption would occur if loss occurred on the live stream during the recovery.
- Corrected a problem with the UME Store when using Microsoft Windows IO Completion Ports that rendered the Store unresponsive to incoming client registration requests. This condition could occur if several clients simultaneously attempted to either register with the Store or initiate keepalive connections.

## 43. Release LBM 3.6.ea2 / UME 3.0ea2 - September 2009

### 43.1. LBM 3.6ea2

#### 43.1.1. New Features

- Added to the LBM Gateway the ability to configure multiple connection sections. This permits a receive-side Gateway to fail over when a connection to the source-side Gateway fails
- Added support for Spectrum to the Java and .NET APIs.

- Added the `<daemon>` section to the `lbmrd` XML configuration file. Child elements are `<port>`, `<log>`, `<activity>`, `<interface>`, `<ttl>` and `<resolver_unicast_receiver_socket_buffer>`. This element can be used to configure the `lbmrd` receive buffer size.

### 43.1.2. Updated Features

- Improved performance of all **29West** products on Microsoft Windows by implementing new build techniques.
- Microsoft has recently provided a security update for the C runtime library (`msvcr80.dll` version: 8.0.50727.4053). In order to avoid potential issues, **29West** recommends installing the latest redistributable C runtime library. See <http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=766a6af7-ec73-40ff-b072-9112bab119c2> for further details.

### 43.1.3. Bug Fixes

- Corrected a problem with Java that could cause a JVM crash if a thread received a small response message (less than 8192 bytes) followed by a large response message (greater than 8192 bytes).
- Updated the .NET version of `lbmpong` to be consistent with the C and Java versions by allowing `lbmpong` to collect data in verbose mode, ignore messages at the head of the stream and calculate the standard deviation, minimum, maximum and average.
- Corrected a problem that occurred if a Microsoft Windows application using completion ports (`fd_management_type (../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE)`) in embedded mode (`operational_mode (../Config/majoroptions.html#CONTEXTOPERATIONALMODE)`) used a separate thread to create an **LBM** Context and that thread ended before the **LBM** Context. For sequential mode, the following warning now appears: `lbm_context_process_events: Completion ports warning; Context creation thread has terminated. Some events may be lost.` In this case, **29West** recommends that you change your application code to keep the context-creating thread around for the lifetime of the context.
- Fixed an issue with SDM Unicode strings in .NET. If a Unicode field was the last field in a packet, the length check produced an assertion because the check was made during the unicode string encoding rather than before the encoding. The length check now occurs prior to string encoding.
- Added the correct channel flags to Spectrum channel messages.
- Modified .NET to pull the source string from the string cache during the `registration_id_ex` callback.
- Corrected a problem with **LBM** receivers which resulted in the receivers containing an incorrect source port for TCP sources.
- Corrected a problem with the creation of a source for a topic which was hashed to the same value as another source topic. This condition caused an infinite loop and has been corrected.
- Corrected a problem with `lbmrcv.c`. This application deleted the receiver before calling `lbmmon_rcv_unmonitor()`. If the monitoring period expired after the delete but before the `unmonitor` call, the following assertion appears: `LOG Level 1: FATAL: failed assertion [rcv->rcv!=NULL] at line`

1576 in ../../../../src/lib/lbm/lbmrcv.c. The receiver delete now occurs after the lbmmon\_rcv\_unmonitor().

- Fixed a Microsoft Windows synchronization problem which resulted in extreme delays if a source object used multiple threads for blocking sends.
- Corrected a problem that could cause a JVM crash if a Java application running in Embedded Mode delivered certain logging callbacks.
- Fixed a problem with Late Join that occurred when recovering old messages from a source. If loss occurred on the live stream during recovery, the source would either stop or delay the delivery of messages to the application and buffer the messages instead, causing unbounded memory consumption. In addition, any messages that were delivered could be delivered out of order.
- Corrected a memory leak that occurred when creating Java `LBMContextStatistics` objects.
- Corrected a problem with the LBT-IPC transport on Microsoft Windows that caused multiple processes that were started in rapid succession to send data to the same shared memory area. This problem produced the following failed assertion at the receiver: `MUL_ASSERT((MUL_SQN_GT(sq, ctrl->stream_high_sqn)));`
- Increased an internal NAK timer constant for LBT-RM receivers to improve loss handling in high-loss cases.

#### 43.1.4. LBM 3.6ea2 Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. 29West is investigating this issue.
- Shutting down a source-side LBM Gateway with Control-C can cause the Gateway to become dead locked. 29West is investigating this issue.
- If a source LBM Gateway was shutdown and then the receiving Gateway was shutdown as quickly as possible while still carrying traffic, a fatal assert occurs. (`MUL_FATAL_ASSERT(Table!=NULL);`) 29West is investigating this issue.
- **LBM Spectrum** does not operate correctly with wildcard receivers. The functions `lbm_wildcard_rcv_subscribe_channel` and `lbm_wildcard_unsubscribe_channel` (as well as `LBMWildcardReceiver.subscribeChannel()` and `LBMWildcardReceiver.unsubscribeChannel()` in Java and .NET) should not be called, and the receiver attributes "null\_channel\_behavior" and "unrecognized\_channel\_behavior" should not be used if wildcard receivers are being used. This issue will be corrected in a later release.

## 43.2. UME 3.0ea2

### 43.2.1. Bug Fixes

- Fixed an issue in .NET and C where the topic index was not included in the source string passed to the user in the **UME** registration callbacks set by `ume_registration_function` and `ume_registration_extended_function`, which are options for `lbm_rcv_topic_attr_setopt()`.



### 43.2.2. Known Issues

- Queue resiliency is not in place with this release. Multiple Queues with the same name are not supported with this release. In addition, Queues that fail and restart will not work with continually functioning sources and receivers.
- Sources must use either **UME** stores or a **UME** queue, but not both at the same time. This will be addressed in a future release.
- When using a **UME** Queue, receivers should not rely upon the message sequence number supplied in the `lbm_msg_t` (or `LBMMessage`) object as the sequence numbering may change in future releases. Applications should utilize the `Message-ID` instead.
- EOS semantics related to Queuing are somewhat complicated depending on the dissemination model being used. New events may be introduced in future releases to handle additional "EOS-like" concerns associated with the use of a **UME** Queue.

## 44. Release LBM 3.5.2 / UME 2.2.2 - August 2009

### 44.1. LBM 3.5.2

#### 44.1.1. Updated Features

- Improved performance of all **29West** products on Microsoft Windows by implementing new build techniques.
- Microsoft has recently provided a security update for the C runtime library (msvcr80.dll version: 8.0.50727.4053). In order to avoid potential issues, **29West** recommends installing the latest redistributable C runtime library. See <http://www.microsoft.com/downloads/details.aspx?displaylang=en&FamilyID=766a6af7-ec73-40ff-b072-9112bab119c2> for further details.

#### 44.1.2. Bug Fixes

- Corrected a problem with Java that could cause a JVM crash if a thread received a small response message (less than 8192 bytes) followed by a large response message (greater than 8192 bytes).
- Fixed a problem with Late Join that occurred when recovering old messages from a source. If loss occurred on the live stream during recovery, the source would either stop or delay the delivery of messages to the application and buffer the messages instead, causing unbounded memory consumption. In addition, any messages that were delivered could be delivered out of order.
- Corrected a memory leak that occurred when creating Java `LBMContextStatistics` objects.
- Corrected a problem with the LBT-IPC transport on Microsoft Windows that caused multiple processes that were started in rapid succession to send data to the same shared memory area. This problem produced the following failed assertion at the receiver: `MUL_ASSERT((MUL_SQN_GT(sqn, ctrl->stream_high_sqn)));`

- Fixed an issue with SDM Unicode strings in .NET. If a Unicode field was the last field in a packet, the length check produced an assertion because the check was made during the unicode string encoding rather than before the encoding. The length check now occurs prior to string encoding.
- Increased an internal NAK timer constant for LBT-RM receivers to improve loss handling in high-loss cases.
- Added a protective check to prevent a segfault when a source sends with monitoring enabled and `resolver_multicast_incoming_address` (`../Config/multicastresolvenetworkoptions.html#CONTEXTRESOLVERMULTICASTINCOMINGADDRESS`) set to `0.0.0.0`.

### 44.1.3. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. 29West is investigating this issue.

## 44.2. UME 2.2.2

### 44.2.1. Bug Fixes

- Fixed a problem when recovering messages from a **UME** store identical to the Late Join problem cited above. If loss occurred on the live stream during recovery, the **UME** store would stop or delay the delivery of messages to the application, buffering the messages and causing unbounded memory consumption. In addition, any messages that were delivered could be delivered out of order.

## 45. Release LBM 3.5.1 / UME 2.2.1 - July 2009

### 45.1. Bug Fixes

- Corrected a socket leak on the receiving side of LBT-RM transports that was introduced in **LBM** 3.3. The socket leak prevented **LBM** from leaving multicast groups when deleting receivers, which could lead to an unexpected growth of multicast group memberships in a process. This fix ensures that the O/S leaves the multicast group associated with the socket, when the last receiver in said group is deleted.

## 46. Release LBM 3.5 / UME 2.2 - July 2009

### 46.1. LBM 3.5

See also *Release LBM 3.5.ea3 / UME 2.2ea1 - May 2009* and *Release LBM 3.5.ea2 - April 2009*.

#### 46.1.1. Bug Fixes

- Corrected a problem with Java and .NET `lbmmon` example applications that resulted in the incorrect display of the source application's process and object IDs.
- Corrected a problem with Java and .NET `lbmmsrc` example applications that caused the following exception when printing IPC statistics: `Unhandled Exception: com.latencybusters.lbm.LBMEInvalException: value at num must be large enough to handle all transport sessions. This exception was caused by an incorrect calculation of the number of transport sessions.`
- **LBM** now ignores any spaces at the end of numeric attributes in **LBM** configuration files.
- Fixed an issue with the wire format for LBM messages where the explicit batching flags were not being correctly converted to network byte order. This resulted in the `LBM_MSG_FLAG_START_BATCH` and `LBM_MSG_FLAG_END_BATCH` flags not being set correctly if sending from a big-endian machine to a little-endian machine, or vice versa. LBM now correctly formats the field in question.

Due to the change in the wire format, mixing **LBM** 3.5 with earlier versions results in incorrectly set batching flags in most cases. This does not impact the behavior of explicit batching on the source side, only the message flags on the receiver side. These flags provide information to the receiving application, and are not used in message delivery, so applications that do not currently use them can mix **LBM** versions without impact.

- Changed the way multiple LBT-IPC processes verify availability of the LBT-IPC transport during Topic Resolution. If two processes check the transport at nearly the same time, the second process will experience a 1 second delay. Multiple processes now verify serially.
- Fixed a problem with calling `lbm_src_flush()` from multiple threads or using `lbm_src_send()` in combination with `lbm_src_flush()` from multiple threads. This problem produced the following assertion. `WARNING: failed assertion [num>0] at line 1382 in ../../../../src/lib/lbm/lbmsock.c`. With this correction, using `lbm_src_send()` in combination with `lbm_src_flush()` from multiple threads does not cause a problem.
- Corrected a problem with setting `*_interface` options from within the **LBM** Configuration File specified by the environment variable `LBM_DEFAULT_CONFIG_FILE`. Setting `*_interface` options in this manner no longer results in an error.

#### 46.1.2. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. 29West is investigating this issue.

## 46.2. UME 2.2

### 46.2.1. Bug Fixes

- Java **UME** receiver applications must now call `LBMMessage.dispose()` when the application is completely finished with the message. Previously, **UME** sent implicit ACKs immediately on return from the user's callback. If a Java **UME** receiver application uses implicit ACKs (the default), **UME** will not send any implicit ACKs unless the application calls `LBMMessage.dispose()`.

## 47. Release LBM 3.6.ea1 / UME 3.0ea1 - July 2009

### 47.1. LBM 3.6ea1

#### 47.1.1. LBM 3.6ea1 New Features

- Added **LBM Spectrum** that allows a source to send topic messages with channel information. A receiver can subscribe (`lbm_rcv_subscribe_channel()` `lbm_wrcv_subscribe_channel()`) to a channel and receive only topic messages sent to that channel. Each channel can use a different receiver callback, if desired. Benefits include less topic resolution traffic since LBM advertises only topics, not channels and in order delivery across channels since all messages are part of the same topic stream.
- Added support for **Adaptive Batching**, which attempts to send messages immediately during periods of low volume and automatically batch messages during periods of higher volume. You enable Adaptive Batching by setting `implicit_batching_type` (`../Config/implicitbatchingoptions.html#SOURCEIMPLICITBATCHINGTYPE`) to `adaptive`.
- **New Monitoring Statistics:** Added monitoring statistics to `lbm_event_queue_stats_t` (`../API/structlbm_event_queue_stats_t_stct.html`) structure to reflect a new event type used internally by **LBM**. The new statistics report on the number of callback events (`callback_events`), total callback events (`callback_events_tot`) and callback event service times (`callback_events_svc_min`, `callback_events_svc_mean` and `callback_events_svc_max`). These statistics are available to the C API (`../API/index.html`), the Java API (`../JavaAPI/html/index.html`) and the .NET API (`../DotNetAPI/doc/Index.html`). The new statistics have also been added to the **Ultra Messaging MIB** and the InterMapper probe files. In addition, all example applications support the callback statistics.
- Added the **Topic Index** to the source string (`lbm_transport_source_info_t` (`../API/structlbm_transport_source_info_t_stct.html`)). **LBM** includes the topic index in source strings supplied through message delivery and new source notification. A receiving application can now identify the source of any message it receives by parsing the source string. New API functions, `lbm_transport_source_parse()` and `lbm_transport_source_format()`, allow you to parse and create source strings. Inclusion of the Topic Index in the source string is the default. You can disable this by setting `source_includes_topic_index` (`../Config/majoroptions.html#CONTEXTSOURCEINCLUDESTOPICINDEX`) to 0 (zero).

## 47.1.2. Bug Fixes

- Removed the warning assertion, `fdm->select_invalid_entries==0`, that could appear during busy periods when using the `select` file descriptor type.
- Corrected a problem that occurred if a NULL topic object was passed to `lbm_hf_src_create`. This resulted in a Segfault. `lbm_hf_src_create` now correctly returns `-1`.
- Changed the default Host ID for Microsoft Windows from `0x00000000` to `0x7fffffff` to improve host name resolution between an **LBM** source running on Microsoft Windows and sending over LBT-IPC to a receiver running on a Linux/Unix machine. The previous default Host ID could produce the following receiver error: LOG Level 5: LBT-IPC: failed to open shared memory (2).
- Corrected a problem with Java and .NET `lbmmon` example applications that resulted in the incorrect display of the source application's process and object IDs.
- Corrected a problem with Java and .NET `lbmmsrc` example applications that caused the following exception when printing IPC statistics: Unhandled Exception: `com.latencybusters.lbm.LBMEInvalException: value at num must be large enough to handle all transport sessions`. This exception was caused by an incorrect calculation of the number of transport sessions. Future versions of **LBM** will remove the requirement for Java and .NET applications to specify a number of statistical sets when creating various **LBM** statistics.
- **LBM** now ignores any spaces at the end of numeric attributes in **LBM** configuration files.
- Fixed an issue with the wire format for LBM messages where the explicit batching flags were not being correctly converted to network byte order. This resulted in the `LBM_MSG_FLAG_START_BATCH` and `LBM_MSG_FLAG_END_BATCH` flags not being set correctly if sending from a big-endian machine to a little-endian machine, or vice versa. **LBM** now correctly formats the field in question.

Due to the change in the wire format, mixing **LBM** 3.5 with earlier versions results in incorrectly set batching flags in most cases. This does not impact the behavior of explicit batching on the source side, only the message flags on the receiver side. These flags provide information to the receiving application, and are not used in message delivery, so applications that do not currently use them can mix **LBM** versions without impact.

- Changed the way multiple LBT-IPC processes verify availability of the LBT-IPC transport during Topic Resolution. If two processes check the transport at nearly the same time, the second process will experience a 1 second delay. Multiple processes now verify serially.
- Fixed a problem with calling `lbm_src_flush()` from multiple threads or using `lbm_src_send()` in combination with `lbm_src_flush()` from multiple threads. This problem produced the following assertion. WARNING: failed assertion [num>0] at line 1382 in `../src/lib/lbm/lbmsock.c`. With this correction, using `lbm_src_send()` in combination with `lbm_src_flush()` from multiple threads does not cause a problem.
- Corrected a problem with setting `*_interface` options from within the **LBM** Configuration File specified by the environment variable `LBM_DEFAULT_CONFIG_FILE`. Setting `*_interface` options in this manner no longer results in an error.

## 47.1.3. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. 29West is investigating this issue.

## 47.2. UME 3.0ea1

### 47.2.1. New Features

- Java **UME** receiver applications must now call `LBMMessage.dispose()` when the application is completely finished with the message. Previously, **UME** sent implicit ACKs immediately on return from the user's callback. If a Java **UME** receiver application uses implicit ACKs (the default), **UME** will not send any implicit ACKs unless the application calls `LBMMessage.dispose()`.
- Added **Queuing** capabilities to **UME** that allows sources to submit messages asynchronously to a queue and permits receivers to retrieve messages from a queue in an entirely different asynchronous manner. **UME** Queuing also supports Once and Only Once (OAOO) delivery and Application Sets that allow you to load balance processing or support multiple processing purposes for single topics. See Ultra Messaging® for the Enterprise ([../UME/index.html](#)) for more information.
- Added support for **Proxy Sources** which allows a **UME** source to request one of the **UME** stores act as a proxy in case the source terminates. In this event, advertisements can continue, existing receivers can continue recovery from a store, and new receivers can register with the stores and receive any messages they have missed. You enable Proxy Sources with the source configuration option, `ume_proxy_source` ([../Config/ultramessagingpersistenceoptions.html#SOURCEUMEPROXYSOURCE](#)).

### 47.2.2. Known Issues

- Queue resiliency is not in place with this release. Multiple Queues with the same name are not supported with this release. In addition, Queues that fail and restart will not work with continually functioning sources and receivers.
- Sources must use either **UME** stores or a **UME** queue, but not both at the same time. This will be addressed in a future release.
- When using a **UME** Queue, receivers should not rely upon the message sequence number supplied in the `lbm_msg_t` (or `LBMMessage`) object as the sequence numbering may change in future releases. Applications should utilize the `Message-ID` instead.
- EOS semantics related to Queuing are somewhat complicated depending on the dissemination model being used. New events may be introduced in future releases to handle additional "EOS-like" concerns associated with the use of a **UME** Queue.

## 48. Release LBM 3.5.ea3 / UME 2.2ea1 - May 2009

### 48.1. LBM 3.5ea3

#### 48.1.1. New Features

- **New Monitoring Statistics:** Added monitoring statistics for the LBT-IPC transport. These statistics are available to the C API (`../API/index.html`), the Java API (`../JavaAPI/html/index.html`) and the .NET API (`../DotNetAPI/doc/Index.html`). The new statistics have also been added to the **Ultra Messaging MIB** and the InterMapper probe files. In addition, all example applications support LBT-IPC transport statistics.

#### 48.1.2. Updated Features

- Changed the internal management of LBT-IPC shared resources. **LBM** can now support thousands of LBT-IPC transport sessions and hundreds of receiving contexts.

As a result, **LBM 3.5ea2** users must clean up old IPC resources before installing **LBM 3.5ea3**. Perform the following actions to properly update to **LBM 3.5ea3**.

1. Use the `lbtipc_resource_manager -reclaim` option to reclaim any orphaned IPC resources.
2. Use the `lbtipc_resource_manager -delete` option to delete the IPC resources database.
3. Install **LBM 3.5ea3**.

#### 48.1.3. Bug Fixes

- Corrected **LBM** socket resource leakage resulting when a socket did not connect to the sender. The socket delete was not completing.
- Modified `lbmmrcv.cs` to only register one immediate callback. The previous implementation registered one callback for each receiver, which caused a single immediate message to be callbacked as many times as there were receivers. Since the callback executes a `dispose()` on the message, it caused a core dump after the first callback was made.
- Corrected a problem with LBT-IPC that caused an infinite loop when subsequent receivers joined an LBT-IPC transport session after the first receiver joined. The problem could happen if the subsequent receivers joined the transport as much as 17 minutes later than the first.
- Corrected a seg fault that occurred when the maximum number of LBT-IPC transport sessions was exceeded.
- Corrected a problem with the way an LBT-IPC receiver handles keepalive messages from an IPC source. This problem caused the following assertion, "WARNING: failed assertion [(MUL\_SQN\_GT(sq, ctrl->stream\_high\_sqn))]" at line 626 in `../src/lib/lbm/lbmrcvdc.c`.

#### 48.1.4. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. 29West is investigating this issue.

### 48.2. UME 2.2ea1

#### 48.2.1. Bug Fixes

- Fixed a problem with the Java and .NET versions of `umesrc` that resulted in the STORE flag being issued instead of the STABLE flag when a quorum of stores is achieved. This problem only occurred when using a Quorum/Consensus UME store configuration.
- In the Java API, changed UME confirmed delivery behavior for receivers to be more in line with C and .NET APIs. The Java API now requires an application to call `LBMMessage.dispose()` when it has finished with an `LBMMessage` object. Previously this had not been required.
- In the Java API, fixed a bug which caused `LBMMessage.isFragment()` to always return false.
- In the Java API, fixed several small UME source-side memory leaks that occurred when using the newly-supported Embedded Mode without an Event Queue. These problems did not affect Sequential Mode with or without an Event Queue nor Embedded Mode with an Event Queue.

## 49. Release LBM 3.5.ea2 - April 2009

### 49.1. LBM 3.5ea2

#### 49.1.1. New Features

- Introduced Interprocess Communication (IPC) LBM transport that allows sources to publish topic messages to a shared memory area managed as a static ring buffer from which receivers can read topic messages. For more information, see `Transport LBT-IPC` (`../Design/lbm-objects.html#TRANSPORT-LBT-IPC`) in `LBM Design Concepts`. See also `Transport LBT-IPC Operation Options` (`../Config/transportlbt-ipcoperationoptions.html`) in the **LBM Configuration Guide**.
- Added support for the `kqueue fd_management_type` (`../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) on Mac OS X.
- Added ability to set `resolver_multicast_incoming_address` (`../Config/multicastresolvenetworkoptions.html#CONTEXTRESOLVERMULTICASTINCOMINGADDRESS`) to `0.0.0.0`, eliminating incoming topic resolution traffic, which is useful for MIM-only contexts.



### 49.1.2. Updated Features

- Significantly improved performance of Java receivers by taking advantage of some newer features of modern Java virtual machines. Embedded mode is now fully supported in Java with the same performance as sequential mode.
- Allowed Intel Core Duo T2600 CPU machines to use high resolution timers instead of `gettimeofday`.

### 49.1.3. Deleted Features

- Removed the finalizer from `LBMTimer` Java class; changes in LBM 3.4 made the finalizer unnecessary.

### 49.1.4. Bug Fixes

- Fixed a problem that caused the string "unknown" to be returned from a call to `lbm_context_attr_str_getopt()` for the context attribute `fd_management_type` (`../Config/majoroptions.html#CONTEXTTFDMANAGEMENTTYPE`) if it was set to either `epoll` or `devpoll`, rather than the string `epoll` or `devpoll`, respectively.
- Fixed a race condition where a .NET source event callback supplied to the `LBMSource()` or `LBMContext.createSource()` constructors had a chance of not being called for an event that occurred before the constructor returned.
- Fixed a problem that prevented `lbmmon` from reporting LBM stats using `lbmsnmp` or UDP transport on .NET. When the `lbmmon src` was created, the LBM transport was used regardless of the value passed in for the transport parameter.
- Fixed a display problem that reversed multicast addresses in warnings regarding LBT-RM transport sessions on little-endian machines. Also removed warnings for `transport_session_maximum_buffer` (`../Config/transporttcpoperationoptions.html#SOURCETRANSPORTSESSIONMAXIMUMBUFFER`) for LBT-RM and LBT-RU transports as this option affects TCP transports only.
- Corrected a problem in LBM that resulted in a Segmentation Fault when using more than one thread to dispatch an event queue. One or more of the dispatching threads was able to access information associated with an event after the memory had been freed by another thread processing another event.
- Corrected a problem in Java and .NET that could result in a failed assertion when using more than one thread to dispatch an event queue. One or more of the dispatching threads was able to access memory that had already been freed as a result of processing the End of Session (EOS) event on a different thread.
- Corrected a problem in LBM that resulted in a Segmentation Fault when performing a delete operation on a source, receiver, or context in parallel with dispatching an event from an event queue. The dispatching thread was able to access information associated with an event after the memory had been freed by the delete operation.
- **LBM** now correctly applies the `resolution_number_of_sources_query_threshold` (`../Config/resolveroperationoptions.html#RECEIVERRESOLUTIONNUMBEROFSOURCESQUERYTHRESHOLD`) option to receivers created in a Context that already has a source for the same topic.

### 49.1.5. Known Issues

- When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. 29West is investigating this issue.

## 50. Release LBM 3.4.1 / UME 2.1.1 - February 2009

### 50.1. LBM 3.4.1

- **Known Issue:** When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. 29West is investigating this issue.
- Fixed an issue with `lbm_src_sendv()` that resulted in the sending of a message twice if you specified an `iovec` of length 1.
- Fixed a race condition in the handling of the semaphore associated with an LBM event queue that would sometimes cause the following harmless, but annoying log message: "WARNING: failed assertion [ev!=NULL]...".
- Added version information to LBM Windows DLLs. You can use Microsoft Windows File Explorer to view a DLL's version number, description and other information.
- Changed `lbmmon_rctl_destroy()` to also destroy the thread created to receive monitor packets in addition to destroying the monitor receive controller. Not destroying this thread lead to invalid memory access if monitor packets were received after the receive controller was destroyed.
- Added new **LBM Configuration Option**, `transport_lbtrm_preactivity_timeout` (`../Config/transportlbt-rmoperationoptions.html#RECEIVERTRANSPORTLBTRMPREACTIVITYTIMEOUT`) which can prevent headloss from pre-LBM 3.3 sources that do not start sending data messages before the receiver's `transport_lbtrm_activity_timeout` (`../Config/transportlbt-rmoperationoptions.html#RECEIVERTRANSPORTLBTRMACTIVITYTIMEOUT`) expires. The preactivity timeout begins when the receiver receives a topic advertisement.
- Fixed various Java and .NET SDM `get_len()` methods to check for a null field type before attempting to determine the length of the object. Also added new Java and .NET constructors for array fields that enable non-null arrays to be created with 0 elements. This matches the behavior of the C API.
- Fixed a problem that caused an LBM receiver using a TCP transport to segfault when the `delivery_control_loss_check_interval` (`../Config/deliverycontroloptions.html#RECEIVERDELIVERYCONTROLLOSSCHECKINTERVAL`) was set to a non-zero value.
- Changed the .NET example application, `lbmresp.cs`, so it does not crash when compiled with optimization on 64-bit machines. The application's functionality was not changed.
- Changed .NET to call `RaiseException` during a Fatal Assert which allows you to use your own exception handler. Previously, .NET called `abort` which would not invoke your exception handler.

- Added internal support for the AIX call `read_real_time()` when timestamping event queue entries. This provides better performance than the default `gettimeofday()` and applies to only AIX.
- Corrected the `LBM*Attributes.GetValue()` calls in .NET to use a platform-specific `size_t` type when getting attribute values. Attribute calls now operate correctly on both 32-bit and 64-bit .NET platforms.

## 50.2. UME 2.1.1

- No UME specific problems were found.

# 51. Release LBM 3.4 / UME 2.1 - February 2009

## 51.1. LBM 3.4

- **Known Issue:** When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. 29West is investigating this issue.
- **Feature Resumption:** String caching for the Java and .NET APIs has been re-enabled. Rather than creating a new string for the topic and source when each message is delivered, **LBM** uses a reference to a cached string that improves performance. In addition, the previously disabled per-source client object has been reintroduced. The object returned from the per-source creation callback is now available in the receiver and per-source deletion callbacks.
- The **LBM** Application Notes section (FAQs, error messages, articles) of the **LBM/UME** documentation set has been moved into the **29West** Knowledgebase (<https://communities.informatica.com/infakb/kbexternal/default.aspx>). We will continue to add articles answering common support questions, new hardware issues, source code examples and more to our new Knowledgebase, which has full text search and ways for you to comment on articles and ask a question of **29West** Support. A link to the current **29West Messaging** documentation set will soon be added. We welcome your comments and hope you find the **29West** Knowledgebase (<https://communities.informatica.com/infakb/kbexternal/default.aspx>) helpful.

### 51.1.1. Monitoring

- **New Monitoring Statistics:** Added Context statistics and Event Queue statistics. Context statistics help you monitor topic resolution activity, along with the number of unknown messages received and the number of sends and responses that were blocked or returned EWOULDBLOCK. Context statistics also contain transport statistics for Multicast Immediate Messaging (MIM) activity and the existing transport statistics for all of the sources or receivers in a context. Event Queue statistics help you monitor the number of events currently on the queue, how long it takes to service them (maximum, minimum and mean service times) and the total number of events for the monitoring period.

Implementing the new context and Event Queue statistics in an existing `lbmmon` statistics receiver application requires the addition of a callback for each statistic type. All new statistics are available in the C API

(../API/index.html), the Java API (../JavaAPI/html/index.html) and the .NET API (../DotNetAPI/doc/Index.html). See also LBM API Functions and Data Structures (../Design/monitoring.html#LBM-API-FUNCTIONS) in LBM Design Concepts. The new statistics have also been added to the **Ultra Messaging MIB** and the InterMapper probe files.

- **Application Change Required:** LBM Version 3.4 has broken source and binary compatibility with monitoring applications that use a monitoring receiver controller. In order for your existing monitoring application to work with LBM Version 3.4, you must change the application code and re-compile. This change, although troublesome now, allows the addition of more statistics in the future without any further application change required.

In the C API (../API/index.html), the method for specifying all statistics callbacks has been changed. Note that this change applies *only* to applications that act as a receiver of statistics. No changes are required for most applications that use the `lbmmon` API to publish statistics. See the `lbmmon` API documentation (../API/lbmmon\_8h.html) for information about the new parameters passed to `lbmmon_rctl_create()`, as well as the new API calls (`lbmmon_rctl_attr_*()`) along with the structures (`lbmmon_rctl_attr_t` and `lbmmon_*_statistics_func_t`). See also Receiving Monitoring Data (../Design/monitoring.html#RECEIVING-MONITORING-DATA) in LBM Design Concepts for code examples that use the new `lbmmon` API calls to create a statistics receiver.

The Java API (../JavaAPI/html/index.html) and .NET API (../DotNetAPI/doc/Index.html) also require a new way to specify statistics callbacks for all statistics. However, callbacks for existing statistics are now deprecated and will continue to work until an undetermined future release.

- Added API functions to reset statistics for all functions that retrieve statistics. For more information, see the C API (../API/index.html), the Java API (../JavaAPI/html/index.html) and the .NET API (../DotNetAPI/doc/Index.html).
- Added a new monitoring function, `lbmmon_sctl_sample_ex()`. This is an extension to the existing `lbmmon_sctl_sample()` call, but allows an AppID string to be specified. This string overrides the AppID string specified in any `lbmmon*_monitor()` calls for this call only, and can be used to allow dynamic identification of the statistics message (via the AppID) to the receiving monitoring application.
- Added the capability to monitor an LBM context automatically by setting LBM Automatic Monitoring Options (../Config/automaticmonitoringoptions.html) in a configuration file or by setting environment variables, which allow you to automatically monitor all LBM contexts. See also Automatic Monitoring (../Design/monitoring.html#AUTOMONITOR) in LBM Design Concepts.
- Added the number of retransmission bytes sent (`rx_bytes_sent`) to the LBT-RM and LBT-RU source statistics. This allows the calculation of a source's retransmission rate as well as the data rate. This new statistic has also been added to the **Ultra Messaging MIB** and the InterMapper probe files. The **Ultra Messaging MIB** has also been restructured to the following order: Index fields (OIDs .1 through .10), auxiliary fields (OIDs .11 through .20) and the normal statistics fields beginning with OID .21.

## 51.1.2. New Features/Functions

- Added new API attribute management functions (`lbm_XXX_attr_create()`, `lbm_XXX_attr_delete()`, `lbm_XXX_attr_dup()`, and `lbm_XXX_attr_default()`). Now attribute structures can only be allocated, manipulated, copied or destroyed via the new API functions. The existing functions, `lbm_XXX_attr_init()` and `lbm_XXX_attr_cleanup()`, still work but have been deprecated. All new development should be done with the new functions. When the deprecated functions are removed in a future release, all `lbm_XXX_attr_t` structures will become completely opaque, and the individual fields will be removed from `lbm.h`.

- Added a new API function `lbm_src_flush()`, which allows you to flush all batched messages from both the implicit and explicit batch buffer. For more information, see the **LBM C API** ([../API/index.html](#)), the **LBM Java API** ([../JavaAPI/html/index.html](#)) and the **LBM .NET API** ([../DotNetAPI/doc/Index.html](#)).
- Added context source event callbacks. **LBM** now delivers source events for any message types that can be sent directly from a context rather than requiring a separate `lbm_src_t` (or `LBMSource`) object. These message types include multicast immediate messages, unicast immediate messages, requests and responses. Currently, the only supported context source event is the wakeup source event type. Also, wakeup source event types for all sources now include a flag field in the event data to indicate what type of source has woken up.
- Added new API functions to retrieve current attribute settings for various **LBM** objects.
- Added C API functions (`lbm*_dump()`) and Java and .NET methods (`dumpAttributeList`) that retrieve an **LBM** object's (context, receiver, Event Queue, etc.) current **LBM** configuration options.
- A new C API function, `lbm_context_delete_ex()`, may be used with an Event Queue cancel callback to provide the application notification that all context-level source events have been processed or purged from any Event Queues. It is similar to the existing `*_delete_ex()` C API functions.

### 51.1.3. Self Describing Messaging

- Added `get_sec()` and `get_usec()` to Java and .NET Self Describing Messaging (SDM). Now you can easily get the values for seconds and microseconds for a timestamp (`LBMSDMRawTimestamp.java` and `LBMSDMRawTimestamp.cs`).
- Added the ability to Java and .NET Self Describing Messaging (SDM) to set the size of the preallocated field array when you create messages with `LBMSDMFieldsAttribute.set_field_prealloc()`. In the C API, this corresponds to the option `field_array_allocation` used with `lbmsdm_msg_attr_str_setopt()`.
- Corrected a problem in the .NET `LBMSDMRawDecimal` class that was misinterpreting the exponent as an unsigned number instead of a signed number when deserializing a message.
- Corrected a problem with Java and .NET Self Describing Messaging (SDM) that prevented the use of strings greater than 32K in length. SDM strings can now be greater than 32K.
- Corrected a problem with Self Describing Messaging (SDM) that occurred when building the serialized form of the message. If the initial estimate of the serialized message length was too small, the output buffer was reallocated. In the cases where the reallocation caused the memory block to be moved, rather than simply extended, this could lead to a write to an invalid memory location.

### 51.1.4. Java

- Updated JNI to catch exceptions in objects created from JNI. If an exception occurs, the exception is logged and the current operation is terminated.
- Corrected a problem with Java timers discovered when using the Java example application, `lbmreq` with Event Queues. Now when an `LBMTimer` object expires and goes out of scope, it will be finalized and collected immediately.
- Fixed a memory corruption problem in the Java API that caused crashes during cleanup of transport sessions for receivers using LBM Event Queues. This problem only occurred on 32-bit systems.

- Fixed race conditions with Java LBMTimers that are constructed without appropriate callbacks being set.

### 51.1.5. Java and .NET

- Added additional null checks in the Java and .NET APIs to prevent null pointer exceptions when handling messages containing empty arrays.
- Fixed an issue with .NET and Java that caused Hot Failover Receiver statistics to be empty. Also fixed .NET and Java `lbmrcv` example programs that printed incorrect values for messages received (`msgs_rcved`) when using the TCP transport and periodic statistics collection.
- Added `LBMContextAttributes.setImmediateMessageCallback()` function in Java and .NET to allow setting a context's topic-less immediate message callback before context creation. This method is now preferred over the older `LBMContext.enableImmediateMessageReceiver()` and `LBMContext.addImmediateMessageReceiver()` methods.
- Updated Java and .NET finalizers to catch unexpected exceptions and dump the stacktrace of the exception to the standard Java API `printStackTrace(stderr)`.
- Updated `LBMReceiver` constructors in Java and .NET to throw an `LBM.EINVAL` exception if they receive an `LBMTopic` object that has already been used in the creation of an `LBMReceiver`. Re-using `LBMTopic` objects in this manner has always been unsupported by **LBM**, but no error was logged previously.

### 51.1.6. Configuration

- Process-wide **LBM** configuration option defaults may now be automatically loaded at startup without code changes by setting the environment variable `LBM_DEFAULT_CONFIG_FILE` to the path (or URL) of an **LBM** configuration file.
- Added the `source_event_function(..Config/majoroptions.html#CONTEXTSOURCEEVENTFUNCTION)` context configuration option as an alternative way to set a context's topic-less immediate messaging callback. The older method, using the `lbm_context_rcv_immediate_msgs` API call, is still supported, but not recommended for new code.
- Added warning message to `rcv_create` functions if the `transport_lbtrm_activity_timeout` (`../Config/transportlbt-rmoperationoptions.html#RECEIVERTRANSPORTLBTRMACTIVITYTIMEOUT`) is less than `transport_lbtrm_nak_generation_interval` (`../Config/transportlbt-rmreliabilityoptions.html#RECEIVERTRANSPORTLBTRMNAKGENERATIONINTERVAL`). The warning states that this condition results in silent data loss if loss occurs within the activity timeout interval prior to the end of the transport session.
- Added two new configuration options, `transport_lbtrm_nak_initial_backoff_interval` (`../Config/transportlbt-rmreliabilityoptions.html#RECEIVERTRANSPORTLBTRMNAKINITIALBACKOFFINTERVAL`) and `mim_nak_initial_backoff_interval` (`../Config/multicastimmediatemessagingreliabilityoptions.html#CONTEXTMIMNAKINITIALBACKOFFINTERVAL`), which control latency due to loss. The default for both options is 50 milliseconds. A value of zero for either option turns on immediate NAKs.

- Added a new Wildcard Receiver option `receiver_create_callback` (`../Config/config.html#WILDCARDRECEIVERRECEIVERCREATECALLBACK`) that allows a callback when a Wildcard Receiver matches a topic and is about to create an actual receiver for the topic. This allows, among other things, an application to modify the receiver attributes to be used in creating the receiver.
- Removed the explicit setting of `request_tcp_port` (`../Config/config.html#CONTEXTREQUESTTCP`) from the C example application, `lbmreq`. `lbmreq` now uses the `request_tcp_port_low` (`../Config/config.html#CONTEXTREQUESTTCPLOW`) and `request_tcp_port_high` (`../Config/config.html#CONTEXTREQUESTTCPHIGH`) range.
- `lbm_config`, `LBM.setConfiguration()` and `lbmaux` functions can now load configuration files from a URL using HTTP or FTP. (e.g. `lbm_config http://www.example.com/lbm.cfg`).
- Corrected a problem with an LBT-RU source with source-side filtering enabled that would cause a segfault if the source option `transport_lbtru_client_activity_timeout` (`../Config/transportlbt-ruoperationoptions.html#SOURCETRANSPORTLBTRUCLIENTACTIVITYTIMEOUT`) was set to less than 10 seconds, and a timeout was detected after a receiver connect.

### 51.1.7. Request/Response

- Fixed an error where held responses could trigger a memory read error. "Held responses" refers to an LBM response or Unicast Immediate Message for which the response object has been retained after the response has been sent, and subsequently, the requestor disconnects. The memory read error occurs upon the next response sent with the response object.
- Upon expiration of the `response_tcp_deletion_timeout`, logic was added to ensure that the response message had been sent. If not, **LBM** restarts the timer to give additional time for the send to complete.
- Corrected a problem on Mac OS X that resulted in a bind failure of the `request_tcp_interface` (`../Config/requestnetworkoptions.html#CONTEXTREQUESTTCPINTERFACE`) or the `response_tcp_interface` (`../Config/responseoperationoptions.html#CONTEXTRESPONSETCPINTERFACE`). The bind failure prevented the initialization of the **LBM** context and produced the message, `could not find open TCP server port in range`. This problem was most prevalent in Java applications, but could also occur in C applications.
- Corrected a problem with sending responses that returned an `LBM_EOP` error, `response channel disconnected unexpectedly` due to the selection of a disconnected channel that could also result in the access of freed memory. Now **LBM** ignores disconnected response channel connection objects to avoid inadvertently referencing freed memory. This correction forces responses sent to a disconnected channel (requester) to re-establish the connection, instead of failing with the `LBM_EOP` error, `response channel disconnected unexpectedly`.

### 51.1.8. General

- Corrected the condition that prevented the flushing of the message batching buffer with the `LBM_MSG_FLUSH` flag after starting a message batch with the `LBM_MSG_START_BATCH` flag. The `LBM_MSG_FLUSH` flag now flushes the buffer, but **LBM** continues to batch messages until it receives the `LBM_MSG_END_BATCH` flag. Additionally, if

- LBM** receives a `LBM_MSG_COMPLETE_BATCH` flag without previously receiving a `LBM_MSG_START_BATCH` flag, it does not report an error.
- **LBM** now supports setting the `SO_RCVBUF` and `SO_SNDBUF` on a Stratus VOS system. If running on a version of VOS that does not support this functionality, **LBM** logs a warning. You may set these parameters via **LBM** configuration options ending in `socket_buffer`.
  - **LBM** now conforms to the POSIX standard when sending a message on a Stratus VOS system. Previously, VOS 17.x did not function properly with **LBM**.
  - **LBM** now allows the `mim_src_deletion_timeout` (`../Config/multicastimmediatemessagingoperationoptions.html#CONTEXTMIMSRCDELETIONTIMEOUT`) option to increase as needed to ensure the last message has completed. Previously, when sending large immediate messages, the message could take longer to send than the `mim_src_deletion_timeout` (`../Config/multicastimmediatemessagingoperationoptions.html#CONTEXTMIMSRCDELETIONTIMEOUT`) default, which resulted in a failure.
  - **LBM** rate limiter now supports values more than 4 gigabits/sec. Previously, rate limits of more than 4 gigabits per second were not supported on 32-bit or 64-bit Microsoft Windows, and were not supported on other 32-bit platforms. Also, configuration file parsing for rate limiter settings has been changed to allow larger (64-bit) values.
  - **LBM** now reports an error if you configure Wildcard Receiver and Event Queue options with non-numeric values. Previously **LBM** would assign a value of zero without any notification.
  - Reactivated the Linger option (linger for xx seconds before closing context) for the Java source code example, `lbmmsg.java`.
  - Corrected how **LBM** Debug enables certain debug levels that resulted in segmentation faults on Solaris.
  - In a few error cases **LBM** aborts an operation and returns without unlocking its mutex locks. **LBM** has been updated to unlock mutex locks before returning from an error.
  - Fixed an issue where a partial message could be delivered after a burst loss notification. As a result of this change, if any portion of a message is lost, the entire message is considered lost.
  - Changed the error message produced when a receiver goes down prior to an application sending a Unicast Immediate Message. Error message now reads, `LBM send error: no connection to target while sending unicast immediate message`.
  - Changed TCP-LB to not send partial messages. When using Bounded Latency TCP and sending large fragmented messages, if **LBM** needs to remove data from the buffer to make room for new data, **LBM** removes all fragments of a message so the receiver does not receive any partial messages.
  - Fixed a problem when using LBT-RU on Microsoft Windows in conjunction with the `wincompport_fd_management_type` (`../Config/config.html#CONTEXTTFDMANAGEMENTTYPE`). Previously, a single receiver disconnecting from a topic caused all other receivers for that topic to stop receiving data.
  - Corrected a problem in the **LBM** Gateway that prevented messages from being forwarded across the Gateway if you used a non-default value for the Unicast topic resolution port in tunnel mode.
  - Corrected a problem with TCP transport that occurred when the `transport_tcp_multiple_receiver_behavior` (`../Config/transporttcpoperationoptions.html#SOURCETRANSPORTTCPMULTIPLERECEIVERBEHAVIOR`) was set to `bounded_latency` or `source_paced`. The "total buffered" counter increased when data was added to the buffer but was not properly decreased when data was removed. Over time, this resulted in the failed assertion



[src->totalbuffered>=len]. This counter supplies the bytes\_buffered source transport TCP statistic and does not impact the performance of LBM.

- Corrected a problem with `lbm_log` that resulted in a system crash when `lbm_log` was called multiple times on machines with multiple interfaces.
- Fixed a condition on Microsoft Windows that caused a blocking MIM send to return immediately without sending any data if it followed a non-blocking send which had returned LBM\_EWOULDBLOCK.
- Corrected a problem with LBT-RU and source-side filtering that caused a deleted and then re-created receiver to connect to the source but not receive any messages.
- Fixed a .NET marshalling issue affecting `LBM.version()`, `LBM.errorMessage()` and `LBMMonitor.errorMessage()` on some 64-bit systems which resulted in crashes when calling these functions or if the .NET API attempted to throw an exception.

## 51.2. UME 2.1

- UME store options, `repository-size-threshold` and `repository-size-limit` may now be safely set up to 4 gigabytes.
- The Java and .NET APIs now properly support UME Quorum/Consensus store failover.

## 51.3. SNMP Agent 1.1

- All new statistics cited above in *Monitoring* have been added to the **Ultra Messaging MIB** and the InterMapper probe files.
- **29West Messaging** now supports the optional LBT-RU session ID. This was also added to the **Ultra Messaging MIB**.
- Fixed a problem that prevented **29West Messaging** from starting due to a properly formed XML configuration file that did not contain the optional `<daemon>` section. Problem only appeared on Microsoft Windows 32-bit systems.

# 52. Release LBM 3.3.9 / UME 2.0.7 - December 2008

## 52.1. LBM 3.3.9

- **Known Issue:** When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. 29West is investigating this issue.

- **Feature Resumption:** String caching for the Java and .NET APIs has been re-enabled. Rather than creating a new string for the topic and source when each message is delivered, **LBM** uses a reference to a cached string which improves performance. In addition, the previously disabled per-source client object has been reintroduced. The object returned from the per-source creation callback is now available in the receiver and per-source deletion callbacks.
- Added support in Java and .NET for changing the callbacks associated with an `LBMReceiverAttributes` object during the `WildcardReceiverCreateCallback`.
- Added performance enhancements to Self Describing Messaging (SDM) which included a new attribute option, `integer_fieldname` which allows you to substitute a non-negative integer for a field name, which enhances field array indexing. In addition, added a new function, `lbmsdm_msg_parse_reuse()`, which allows parsing a message into an existing `lbmsdm_msg_t` structure. SDM clears the message before parsing, removing all existing fields from the message. See the API documentation for more information ( C API ([../API/index.html](#)), Java API ([../JavaAPI/html/index.html](#)) or .NET API ([../DotNetAPI/doc/Index.html](#))).
- **LBM** now only supports 64-bit systems on MAC OS X. **LBM** no longer supports 32-bit MAC OS X systems.
- Added a log warning for Microsoft Windows users to advise on possible loss from the combination of high settings for the rate limiter and low source side buffer settings used with LBT-RM or LBT-RU.
- Introduced code to properly avoid allowing callback objects from being garbage collected prematurely. New code also aids in the implementation of string caching. For Java and .NET only.
- Fixed a case where an internal reference count could become negative when using wildcard receivers. This resulted in the wrong receiver attributes being used in cases where multiple receivers for the same topic were being used by a wildcard receiver whose pattern matched the topic in question.
- Fixed a problem that caused the `WildcardReceiverCreateCallback` to not be called in instances where a previous receiver for that topic already existed.
- Added a `WildcardReceiverDeleteCallback`. This callback is symmetric with the `WildcardReceiverCreateCallback`, and provides notification when one of the underlying receivers created by a wildcard receiver is deleted.
- Resolved an issue in .NET with a `NullReferenceException` that was generated when statistics were received by an `LBMMonitorReceiver`. `LBMStatistics` now correctly handles the timestamp received from `lbmmon`.
- **LBM** now prints a warning message every time a source or receiver joins a transport session that has settings for certain attributes different from what the source or receiver has configured. As always, the source or receiver must use the existing transport session's attributes.
- Resolved an issue with calling `lbm_rcv_topic_lookup` without a corresponding `lbm_rcv_create` that caused a double free upon context deletion. **LBM** now correctly disposes of the topic upon context deletion.
- Fixed a deadlock issue on Microsoft Windows caused by creating attribute objects on multiple threads simultaneously.
- Corrected a utility queue removal function (`utl_queue_remove_timedwait()`). The function held the queue lock too long, causing one thread to monopolize the lock, preventing another thread from accessing the queue. This problem only occurred on Microsoft Windows.
- Reverted the `LBMTimer.cancel()` method in the Java API to not take arguments. This corrects an inadvertent change in **LBM** 3.3.7 that required developers to change their applications to call `LBMTimer.cancel(false)` instead of `LBMTimer.cancel()`.

- Fixed an issue when using the *devpoll* `fd_management_type` on Solaris which could cause a fatal assertion if the application was interrupted with a signal.

## 52.2. UME 2.0.7

- Fixed an issue regarding an unexpected interaction between the `WildcardReceiverCreateCallback` and the recovery sequence number callback that resulted in a segfault in Java. This error occurred when using **UME** and registering for the `WildcardReceiverCreateCallback`.

## 53. Release LBM 3.3.8 / UME 2.0.6 - October 2008

### 53.1. LBM 3.3.8

- **Known Issue:** When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. 29West is investigating this issue.
- **Feature Suspension:** String caching for the Java and .NET APIs has been temporarily disabled due to unstable operation. In addition, for the `setSourceNotificationCallbacks` in Java and .NET (`LBMReceiverAttributes`), LBM temporarily ignores the value returned from the source creation callback. Relevant calls to retrieve this object will return NULL. String caching and full `setSourceNotificationCallbacks` functionality will be re-enabled in a later release.
- Added the `Source_Paced` value to the `transport_tcp_multiple_receiver_behavior` (`../Config/transporttcpoperationoptions.html#SOURCETRANSPORTTCPMULTIPLERECEIVERBEHAVIOR`) option. TCP senders can now send as fast as they can without regard to the speed of the receivers. This may cause lost messages.
- Fixed memory corruption problem in Java API which caused crashes during cleanup of transport sessions on 32-bit systems for receivers using LBM Event Queues.
- Corrected a fatal assertion ("`map!=NULL`") on an LBT-RU source caused when using Source Side Filtering (SSF) that resulted from deleting a receiver which had not completely connected to the corresponding source.
- Fixed a race condition in the Java API that occurred when a context is closed that allowed `LBMTimers` to be constructed without appropriate callbacks.
- Fixed a race condition in the .NET API with `LBMContext.scheduleTimer()` which could result in callbacks not being executed for very short duration timers because the timers would expire before their callbacks had been registered. Timer callbacks are now registered before the timer is actually scheduled.
- Fixed a race condition in the .NET API which could cause crashes or deadlocks when calling `LBMContext.close()`.
- Fixed an issue in the .NET API caused by marshalling data between managed and native code. Data was being lost after being passed down to native code, which lead to deadlocks in very rare cases.
- Fixed a very small memory leak (~40 bytes) when deleting an LBM context on Microsoft Windows. A mutex used for MIM was not being freed.

- Reverted `LBMTimer.cancel()` method in Java API to take no arguments. An overload of the `cancel()` method which took a boolean argument was inadvertently added in **LBM 3.3.7** in place of the original method. This meant that users of 3.3.7 had to call `LBMTimer.cancel(false)` instead of `LBMTimer.cancel()`.
- Added **LBM** log warnings when connection disconnects or connection deletion occurs when data exists in the send queue.
- Added the new LBM Configuration option, `transport_tcp_activity_method` (`./Config/transporttcpoperationoptions.html#RECEIVERTRANSPORTTCPACTIVITYMETHOD`) to support the ability for receivers on TCP transports to use TCP keepalive support in the Operating System for determining transport session liveness.
- Corrected a problem with the LBM Gateway running on Microsoft Windows that prevented traffic from being sent across a filtering gateway.
- Corrected a problem with Multicast Immediate Messaging (MIM) sends when the rate limiter is exceeded so that it no longer indefinitely blocks.

## 53.2. UME 2.0.6

- No specific UME problems were discovered.

# 54. Release LBM 3.3.7 / UME 2.0.5 - September 2008

## 54.1. LBM 3.3.7

- **Known Issue:** When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. 29West is investigating this issue.
- Removed source IP address from LBT-RU receiver filtering criteria to avoid message loss caused by messages sent from the wrong interface.
- Implemented mechanisms to preallocate a given number of SDM fields to improve efficiency (Java and .NET).
- Added context `request_tcp_bind_request_port` (`./Config/requestnetworkoptions.html#CONTEXTREQUESTTCPBINDREQUESTPORT`) configuration option to allow a context to be created that does not bind the request port. This option can be used when Request/Response, Late Join, Unicast immediate reception, or UME capabilities are not needed.
- Fixed problem that prevented source advertisements from being sent after any sort of advertising socket send error.
- Allow the receiver `resolution_number_of_sources_query_threshold` (`./Config/resolveroperationoptions.html#RECEIVERRESOLUTIONNUMBEROFSOURCESQUERYTHRESHOLD`) to be set to zero to effectively disable the sending of topic queries from a particular receiver.
- Added Java and .NET `LBMContext.scheduleTimer()` methods that hold a reference to the `LBMTimer` object by the `LBMContext` to ensure that the `LBMTimer` object does not get garbage-collected prior to expiration.

- Provided new application callback function for wildcard receivers that will be called when the messaging framework discovers a source publishing on a topic matching a receiver's wildcard pattern. This functionality also supports application logic to configure attributes specific to the underlying new receiver object for that topic.
- Removed `lbmcssdm.dll` dependency on `lbmcs.dll`.
- Created `LBMSSDM` jar containing only SDM classes.
- Added `LBMReceiver.receiverCount()` method to return the number of callbacks registered to a given receiver.
- Added `LBMSSDMRawTimestamp.get_sec()` and `.get_usec` methods to `.NET` and Java SDM.
- Fixed looping condition caused by internal recursive use of SDM `LBMSSDMArrayMessage.get_len()` method.
- Implemented `delivery_control_maximum_total_map_entries` attribute to limit the amount of memory utilized by the receiver delivery controller. This was left undone by reimplementing of the receiver delivery controller in a prior release.
- Fixed `NullPointerException` that occurs when `.NET` `LBMMessage.topicName()` or `LBMMessage.source()` called in EOS handling code.
- Fixed `/dev/poll` usage problem which causes a fatal assertion: "[fdm->devpoll\_last\_result>=0] at line 2121 in ../../../../src/lib/lbm/lbmfd.c"
- Fixed `transport_[lbtrm | lbtru]_source_socket_buffer` implementation to pick-up OS default if OS default is larger (as documented).
- Fixed problem related to copying of SDM blobs (Java and `.NET` only).
- Improved performance of Java and `.NET` SDM string handling.
- Implemented additional debug facility to use `LBM_DEBUG_ASSERTFILE` environment variable to direct dump of in-core debug data after fatal assertion.
- Fixed formatting of `UInt64` (Java) and `float/double` (`.NET`) SDM fields.
- Fixed cloning of SDM message arrays (Java and `.NET` only)
- Fixed `LBM.SRC_EVENT_WAKEUP` under Java which was previously not being delivered after an `LBMEWouldBlockException`.
- Fixed handling of large numbers of transports sessions to avoid segmentation fault.
- Provided additional synchronization to avoid unexpected exception related to the use of an Event Queue with Java and `.NET` string caching (problem originated in **LBM 3.3.6**).
- Prevented bogus topic query from being emitted from `.NET` wildcard receivers even when wildcard topic queries are disabled (problem originated in **LBM 3.3.6**)
- When registering a file descriptor using `epoll`, previous releases would fatally assert with the message: `fatal assertion (epoll_err==0) line 1034 lbmfd.c in function lbm_fd_register` whenever an error was returned. This has been changed to either print a log message when `epoll` returns `EBADF` (Tried to register a bad file descriptor) or `ENOENT` (Tried to perform an operation on a socket that is already closed). All other error types will still fatally assert.
- Fixed two memory leaks that occurred during context deletion. The first leak was 72 bytes and occurred after deleting a context that had been used with an **LBM** source. The second leak varied in size depending on the topics cached by the context. It occurred after a receiver was deleted and then the associated context was deleted before

the current `resolver_query_interval` (`../Config/resolveroperationoptions.html#CONTEXTRESOLVERQUERYINTERVA`) had expired.

## 54.2. UME 2.0.5

- Added support for **UME** store daemon on Solaris 5.10.
- Fixed a problem with source re-registration that caused the source to start sending with the wrong sequence number. This problem occurred when a source had not sent any data before it re-registered with the store. The source misinterpreted the registration information it received from the store and did not start sending with sequence number zero. Receivers would appear deaf during long `retransmit request generation intervals` (`../Config/latejoinoptions.html#RECEIVERRETRANSMITREQUESTGENERATIONINTERVAL`) waiting for sequence number zero.
- Corrected a problem that prevented the Solaris UME store from cleaning up cache and state files for inactive sources and receivers due to failed TCP keepalive connections from the Solaris UME store to clients.

## 55. Release LBM 3.3.6 / UME 2.0.4 - August 2008

- **Known Issue:** When using Event Queues with the Java API on Mac OS X kernel 9.4, core dumps have occurred. Mac OS X kernel versions prior to 9.4 have not produced this behavior. 29West is investigating this issue.
- Added `LBMMMessage` class source and topic string caching to improve memory utilization dynamics (.NET and Java).
- Added new extended Hot Failover send capability in order to receive sequence number information to C, .NET, and Java APIs.
- Fixed memory corruption problem related to the use of `select` file descriptor management when registering a large number of file descriptors at the same time.
- Simplified `LBMMMessage` creation within the Java API layer to avoid holding a reference to a C layer message for certain message types (e.g., requests).
- Fixed Java API access violation exception that appeared when performing a send using an `LBMSourceSendExInfo` object without setting the client object information.
- Fixed `segfault` caused by using `lbmsdm_msg_clear()`.
- Fixed `lbm_fd_cancel` processing of file descriptors managed by `epoll` to avoid fatal assertion.
- Implemented .NET string caching.
- Implemented .NET data marshalling performance enhancements.
- Added error messages to `lbmrd` that indicate certain configuration file errors.
- The .NET API now allows the reuse of `LBMSourceSendExInfo`.
- Cleaned-up the Microsoft Windows build to ensure that the `lbm.dll` has dependencies on only one Microsoft runtime DLL to avoid problems caused by attempting to use different memory heaps.

- Made the maximum datagram size configurable. See `context transport_datagram_max_size` ([./Config/majoroptions.html#CONTEXTTRANSPORTDATAGRAMMAXSIZE](#)).
- 
- Fixed Gateway fatal assertion (`[Source->topic->source!=NULL]` at line 769 in `../../../../src/lbmgwd/lbmgwod.c`) resulting from an attempt to use the outbound side before it is completely initialized.
- Prevent premature garbage collection of `LBMTimer` .NET objects.
- Fixed problem causing sources to advertise topics in response to a non-matching wildcard query.
- Fixed condition causing a fatal assertion (`[epoll_err==0]` at line 1494 in `../../../../src/lib/lbm/lbmfds.c`) when a source is deleted on a non-context thread.
- Fixed Java response support for topicless request messages.
- Made the construction of SDM (Self Describing Messaging) name tree optional to allow better performance.
- Fixed SDM deserialization of nested messages.
- Fixed .NET API structure alignment problem during **UME** receiver registration that sometimes caused an `ArgumentOutOfRangeException`.
- Receiver registrations are now ignored at the store until a topic resolution advertisement has been received from the corresponding source. This avoids a window that causes the receiver to request invalid retransmissions from the store.
- Fixed receiver's incorrect handling of the retransmission request target for round-robin store failover that manifests itself when a store fails during recovery of that receiver.

## 56. Release LBM 3.3.5 / UME 2.0.3 - June 2008

- Fixed an exception caused when using `.NET LBMReceiverAttributes.setRegistrationId()` callback on 64-bit Microsoft Windows.
- Fixed segfault related to Self Describing Messaging (SDM) that occurred when adding message fields after calling `lbmsdm_msg_clear()`.
- Fixed Microsoft Windows build to ensure that memory allocation/deallocation does not occur across DLL boundaries.
- Added the configuration option `context transport_datagram_max_size` to control the maximum datagram size generated by **LBM**.
- Fixed address exception problem caused by the use of Microsoft Windows Completion Ports mapped to 64-bit addressable regions of memory.

## 57. Release LBM 3.3.4 / UME 2.0.2 - April 2008

- Fixed a struct alignment problem affecting only 64-bit .NET **UME** receivers that could cause a `System.ArgumentOutOfRangeException` upon calling the set registration ID callback.

## 58. Release LBM 3.3.3 / UME 2.0.1 - April 2008

- **Known Issue:** Java receivers using ordered delivery and experiencing excessive loss may run out of memory and fail if an adequate JVM heap size is not used. For lossy receivers experiencing memory problems, maximum JVM heap size should be increased.
- Fixed a memory corruption problem that occurred when registering a large number of file descriptors and using `select` as the `fd_management_type`.
- Removed the limitation of 64 receivers per source for the **UME** store daemon.
- Fixed a problem with the **LBM** timer normalization code that caused occasional long processing delays for large microsecond parameter values.
- Added `lbm_hf_src_send_ex()` function to the C API and corresponding `LBMHotFailoverSource.send()` method to the .NET API.
- Fixed memory corruption problem on Windows UME store daemon caused by an uninitialized variable.
- Fixed a problem that prevented TSNI (Topic Sequence Number Information) from being sent when the context's `resolver_active_threshold` is set to 0.
- UME store daemon's web monitor must now be explicitly configured in the store daemon's XML Configuration File.
- Added in-memory debug log capability which can be flushed with `lbm_debug_dump`.

## 59. Release LBM 3.3.2 / UME 2.0 - March 2008

- Changed Java layer to properly log warnings as warnings and not configuration messages.
- Changed Java layer to stop erroneously propagating error codes returned from `processEvents` to the application.
- Added the `retransmit_message_caching_proximity` configuration option to allow a receiver to control the caching of new messages during a Late Join or **UME** recovery.
- Fixed a segmentation fault that could occur if a TCP source is deleted while there is data on the socket.
- Corrected the permanent blocking of a sender if one of two receivers disconnected while the source buffers were full.
- Default value for the configuration option, `transport_tcp_activity_timeout`, was changed to zero. To use the timer, you must configure the option with a value in milliseconds.



- The default value for `mim_ordered_delivery` has been changed to 1, ordered delivery. Previously, the option defaulted to arrival order.
- Switched from Soft References to Weak References in Java to clean up request messages more efficiently.
- Added support in the **LBM** Gateway for MIM fragmentation/reassembly.
- The **UME** source controller now uses the `request_tcp_interface` address as a Persistence Registration response address if it is specified. Otherwise it uses the default multicast interface address.
- All system messages generated during system initialization are now routed to the logging callback function.

## 60. Release LBM 3.3.1 / UME 2.0 - February 2008

See also *Release LBM 3.3 / UME 2.0 - February 2008*, *Release LBM 3.3EA3 / UME 2.0EA2 2008-1-11*, *Release LBM 3.3EA2 / UME 2.0EA1 2007-12-19* and *Release LBM 3.3EA1 2007-11-15*.

- Fixed potential concurrent modification exception that may appear when adding certain callbacks in .NET.
- Reinstated previously deprecated constructors required for the proper specification of **LBMReceiver** callbacks for durable subscribers and **LBM** Late Joins.

## 61. Release LBM 3.3 / UME 2.0 - February 2008

See also *Release LBM 3.3EA3 / UME 2.0EA2 2008-1-11*, *Release LBM 3.3EA2 / UME 2.0EA1 2007-12-19* and *Release LBM 3.3EA1 2007-11-15*.

- Changed the default configuration settings for all 23 port configuration parameters in **LBM**. Each port default was moved by 10,000. For example, `mim_destination_port` moved from 4401 to 14401. The **LBM** code will use the original locations with the definition of the "LBM\_USE\_ORIG\_DEFAULT\_PORTS" environment variable (value not pertinent).
- Implemented Topic Sequence Number Information (TSNI) messages. These will be received from **LBTRM/LBTRU** sources that have been idle for more than the `transport_topic_sequence_number_info_interval`. Older receivers won't understand TSNI messages and will display the log message, "LOG Level 5: LBMC cntl unknown next header 20, ignoring header". This may be safely ignored.
- Added support for multiple **LBM** transport configuration parameters to be set through the `lbmmon` API.
- Added log message identifying the presence of multiple network interfaces before a specific resolver interface has been specified.
- Named all **LBM** data structs to support forward declaration.
- Added propagation of Java exceptions incurred during **LBM** callbacks back to JVM as **LBMApplication** exceptions.
- Changed **LBM** and **UME** to ignore insignificant whitespace in XML-based configuration files.
- Added file name and line number to memory allocation failure messages.

- Added better detection of invalid **LBM** configuration attribute names and values.
- Changed **UME** to flag instances where a store is not seeing topic advertisements from a source.
- Fixed bug in Java and .NET SDM LBMSDMField.sameBaseType() method which always returned true.
- Removed the fatal assertion on invalid sequence number checkin **UME** store that appeared when an unrecoverable loss burst message is received.
- Added failover optimization to **UME** store initialization.
- Fixed a race condition in the **UME** store with respect to source deletion and pending AIOs that could cause a crash.
- Added throttling of unrecoverable loss messages in the **UME** store log.

## 62. Release LBM 3.3EA3 / UME 2.0EA2 2008-1-11

- Added the receiver configuration attribute (transport\_lbtru\_maximum\_connect\_attempts) to enable LBT-RU receivers to detect and clean-up after detection of a dead source.
- Added Topic Sequence Number Information messages to enable faster detection of loss for intermittent senders.
- Added use of shared sockets for LBT-RM receivers listening to different sessions using the same multicast address and port.
- Added gather send support to C API (lbm\_src\_send/lbm\_src\_send\_ex).
- Added optional session IDs to LBT-RU transports to better distinguish between different LBT-RU sessions originating from the same source.
- Changed receiver setup to ignore certain socket errors, preventing the receiver from being created in an inconsistent state which can cause fatal assertion logic to appear when the receiver is eventually deleted.
- Fixed initialization of Java LBMReceiverStatistics and LBMSourceStatistics objects originating from a monitoring callback.
- Fixed calculation of LBT-RM source statistics transmission window size.
- Fixed improper locking in receiver statistics retrieval code that under certain circumstances could allow an assertion (\*num>=ctx-&num\_rcv\_demuxs) and/or subsequent corruption problem.
- Changed maximum topic name length to 246 bytes to avoid corruption problems with multicast immediate messages.
- Corrected reallocation of file descriptor map used in select file descriptor management to avoid assert (mapentry->exceptcb!=NULL) and corruption.
- Added **UME** store configuration attribute (receiver-new-registration-rollback) to allow the store to better control late join behavior for new receivers.
- Increased minimum disk file limit for the **UME** store to 196992 bytes.

## 63. Release LBM 3.3EA2 / UME 2.0EA1 2007-12-19

- Fixed segmentation fault resulting from a previous change to minimize duplicate recovery messages during a **UME** source restart.
- Allow **UME** sources to re-register with store using different topic.
- Added ability for **UME** receivers to specify the message sequence number with which to resume reception of data.
- Added support for explicit **UME** ACKing instead of relying upon **UME** to automatically ACK messages when they are disposed.
- Fixed condition that caused the "[num\_rcvs==src\_state->num\_rcvs]" warning assertion in stored/umestate.c
- Modified **UME** to continue normal processing after catching EIO error that caused the following warning: "[WARNING]: handle events returned error 5 [umestore\_state\_new\_rcv msync: (5) Input/output error]". **UME** prints a log message if the store is located on a network file system.
- Fixed the condition that caused the "[src->repo!=NULL]" fatal assertion in stored/umetopic.c during store source deletion.
- Fixed a condition that prevented the reporting of stability notices to a **UME** source from a secondary store (after failover).
- Fixed a condition that prevented the reporting of stability notices after a **UME** source is restarted.
- Fixed a condition that prevented a store from performing late join when starting up, which in turn, caused unrecoverable loss by the store.
- Corrected logic that caused a segmentation fault during ACK cleanup on a **UME** receiver.
- Modified **UME** to set an error condition when a **UME** source attempts to re-register a pre-existing registration ID with a different topic.
- Modified the **UME** store web monitor to avoid static buffer restrictions based on the on size of response buffer.
- Added **UME** store keepalives to proactively head-off source re-registrations due to a seemingly unresponsive store during periods of unrecoverable loss.
- Fixed umesrc example code to handle sequence number 0 for reclamation callbacks.
- Added optional setting for socket binds on Windows to prevent binding of in-use ports on Windows.
- Added the source configuration setting, `ume_message_stability_notification`, to allow **UME** message stability notifications to be controlled separately from confirmed delivery notification.
- Added **UME** configuration options to control how **UME** sources and receivers select sequence numbers on start-up.
- Replaced **UME** store and late join source hash tables with Alternating Skip Lists to avoid the need to configure table sizes.
- Added **UME** configuration control to allow store to associate CRC checksums with stored messages.
- Fixed `lbmsdm.h` to support changing a field type.
- Established better handling of messages to minimize latency when messages are dropped due to send buffer overflow.
- Modified **UME** to send ACKs directly to **UME** sources when the **UME** 2.0 store is in use.

- Changed **UME** to persist receiver location information within **UME** store to avoid deleting quiescent receivers.
- Fixed **UME** store web monitor to properly update after a source restarts.
- Ported the **UME** store daemon to run on Microsoft Windows.
- Fixed `txw_msgs` field in the LBT-RM source statistics set to correctly represent the number of messages in the transmission window. Also fixed `bytes_reved` field in the LBT-RM and LBT-RU statistics to be consistent with the source statistics (which includes the header size).
- Modified LBN to interpret a zero (0) setting of `transport_lbtrm_receiver_socket_buffer` to indicate the use of the system default.
- Improved the efficiency of source deletion.
- Named all **LBM** external structs to allow pre-declaration of **LBM** structures.
- Added Hot Failover configuration setting to allow applications to receive (flagged) duplicate messages.
- Added a message field containing a source identifying token that can be set by the application for each new source and used for more efficient message dispatching.

## 64. Release LBM 3.2.3 / UME 1.2.1 2007-11-27

- Corrected improper locking of the Retransmission Queue which caused the following assertion in `lbtrm_txw_purge_rdata(): [bvec!=NULL]` at line 245 in `../src/lib/lbm/lbtrm/lbtrmtxw.c`
- Corrected logic in .NET `LBMMessage` class to prevent calling underlying C message deletion function with NULL pointer.
- Corrected problem with the premature deletion of the socket support in a context thread that caused the following assertion `[1e40:1f98|2140.696]:failed assertion [lookup_mapentry!=NULL]` at line 2330 in `lbm\lbmfd.c`
- Corrected problem with sources using Windows completion ports and sending low message counts over TCP that caused the source to hang and restart after receivers are stopped.
- Corrected condition that prevented receivers from getting messages due to TCP connections closing without proper end of session activity.
- Fixed logic that prevented a **UME** receiver configured with `late_join` set to 1 and `ume_store` set to 0 from requesting initial retransmissions.
- Corrected problem with the handling of disallowed topics in the **LBM** Gateway using filter mode.
- Fixed problem with **LBM** Gateway monitoring that resulted in empty application-ID field.
- Addressed multiple restart issues by changing the initial topic index value to a random value for TCP, LBT-RM, and LBT-RU. For TCP and LBT-RU, the initial value is random at context creation. For LBT-RM, each initial transport creation uses a random value.
- Addressed thread safety issue by synchronizing access to callback lists within the .NET API.
- Corrected problems with duplicate sources for the same topic on either side of the **LBM** Gateway which resulted in inaccurate topic state information after receivers are stopped. The Gateway will now ignore Leave Topic messages for topics that do not have a proxy source.

- Added a notification from `lbm_context_delete` to the context thread to prevent an unnecessary 500ms delay when no events are occurring on the context that is being deleted.
- Corrected problems with non-blocking TCP connections on Solaris that caused either message loss or the following assertion: FATAL: failed assertion [`dmux!=NULL`] at line 1083 in `../../../../src/lib/lbm/lbmrcv.c`.

## 65. Release LBM 3.3EA1 2007-11-15

- Corrected improper locking of the Retransmission Queue which caused the following assertion in `lbtrm_txw_purge_rdata()`: [`bvec!=NULL`] at line 245 in `../../../../src/lib/lbm/lbtrm/lbtrmtxw.c`
- Corrected problem with the premature deletion of the socket support in a context thread that caused the following assertion [`1e40:1f9812140.696`]:failed assertion [`lookup_mapentry!=NULL`] at line 2330 in `lbn\lbnmfd.c`
- Corrected problem with sources using Windows completion ports and sending low message counts over TCP that caused the source to hang and restart after receivers are stopped.
- Corrected condition that prevented receivers from getting messages due to TCP connections closing without proper end of session activity.
- Ensure that C API functions do set return object pointer until corresponding object has been completely created.
- Added debug flag to log a message every time an **LBM** API function is called.
- Added the logging of gateway (`lbgwd`) configuration data once the configuration has been processed, but prior to the start of inbound and outbound components.
- Add generation count to timer IDs to prevent accidental cancellation of a timer that has already expired.
- Improve efficiency of timer cancellation to avoid searching for a timer on an event queue if the timer has not yet expired.
- Fixed problem with SDM that prevented the re-addition of fields to the name tree by ensuring the deletion of field entries from the name tree after removal.
- Added `lbn_debug_mask()` and `lbn_debug_filename()` to the **LBM** API which allows an application to initialize the debug settings that previously could only be set via environmental variables.
- Corrected problem with the handling of disallowed topics in the **LBM** Gateway using filter mode.
- Fixed problem with **LBM** Gateway monitoring that resulted in empty application-ID field.
- Added support for recurring timers with new API function, `lbn_schedule_timer_recurring()`.
- Corrected denial of service problems as the result of receiving unusual messages sent to the topic resolution multicast group.
- Addressed multiple restart issues by changing the initial topic index value to a random value for TCP, LBT-RM, and LBT-RU. For TCP and LBT-RU, the initial value is random at context creation. For LBT-RM, each initial transport creation uses a random value.
- Addressed thread safety issue by synchronizing access to callback lists within the .NET API.
- Corrected problems with duplicate sources for the same topic on either side of the **LBM** Gateway which resulted in inaccurate topic state information after receivers are stopped. The Gateway will now ignore Leave Topic messages for topics that do not have a proxy source.

- Corrected problem with calls to `lbm_errmsg()` and `lbm_errnum()` that produced error messages consisting of the text "no error". Uninitialized error strings will now display "no error message set".
- Added a notification from `lbm_context_delete` to the context thread to prevent an unnecessary 500ms delay when no events are occurring on the context that is being deleted.
- Added support for Late Join in **LBM** independent of **UME**.
- Corrected problems with non-blocking TCP connections on Solaris that caused either message loss or the following assertion: FATAL: failed assertion [dmux!=NULL] at line 1083 in `.././././src/lib/lbm/lbmrcv.c`.

## 66. Release UME 1.2.1EA1 2007-10-30

- Addressed multiple restart issues by changing the initial topic index value to a random value for TCP, LBT-RM, and LBT-RU. For TCP and LBT-RU, the initial value is random at context creation. For LBT-RM, each initial transport creation uses a random value.
- Improved efficiency of large **UME** recovery windows with a redesign of the receiver delivery controller.
- Fixed a condition where sources restart during a long **UME** recovery with LBT-RM by deleting duplicate Registration IDs from previous transport sessions before they can be used for new sessions.
- Modified `umestore` to set Initial Sequence Numbers only if no topic resolution information exists which addresses random store failures. Also fixed the display of the store index for updated stores on the receivers.
- Turn resolver caching off for **UME** store to stop assertion caused by maintaining previous transport information. The assertion was: `[MUL_SQN_GTE(sqncntrl->order.expected_sqncntrl)]` at line 581 in `.././././src/lib/lbm/lbmrcvdc.c`.
- Added check to print warning when `retention_size_limit` for Late Join and **UME** is less than the maximum message size (64KB) and ensure that at least one message is retained in either place in this situation.
- Adjust 90% computational logic for checking the size of the repository-size-limit relative to repository-disk-file-size-limit to avoid false failure indications.
- Fixed condition where **UME** restarted receivers may ignore the registration ID callback and use 0 as the requested registration ID.
- Fixed order of operations in handling re-registrations in store to avoid spurious "PREG received for unknown source/receiver" warnings.
- Fixed condition where a retransmission prematurely turns off generation of more retransmissions, causing **UME** receivers not finished with retransmissions to experience unrecoverable loss.
- Changed Hot Failover event processing to pass through important events previously being ignored (e.g., registration status).
- Corrected blocked sender recovery logic in `umesrc` example so that it passes the callback object needed for proper handling of stability notices.

## 67. Release LBM 3.2.2 2007-10-08

- Fixed handling of TCP connection refused errors on Solaris in rare circumstances when TCP sources were deleted before corresponding receivers
- Corrected logic in context deletion code to avoid unnecessary 500 millisecond delay

## 68. Release LBM 3.2.1 / UME 1.2 2007-09-06

- Allowed greater than 30 minute **UME** activity timeout for **UME** stores running on 32-bit machines, in order to prevent the overflow of a time variable that would cause source and receiver state to be cleaned up prematurely by the store daemon.
- Added support for `ume_retransmit_request_outstanding_maximum` configuration setting, to control the number of outstanding retransmission requests at any one time.
- Prevent crash of **UME** store daemon, caused by reception of web monitor requests greater than 1024 bytes in length.
- Added support for detecting connection failures on Windows clients, so that sources and receivers can reconnect properly to restarted **UME** store.
- Added work-around to Windows problem that prevents a registration ID of 0 from being converted into a unique registration ID.
- Added suppression of logging superfluous **UME** keep alive messages for unknown **UME** sources and receivers.
- Added `-m` (message rate) and `-f` (unstabilized message backlog) options to `umesrc` to enable better rate control for `umesrc`.
- Fixed duplicate ACK condition, resulting when **UME** receivers receive fragmented messages, by properly clearing the `umeack` information in the individual fragments before deleting the messages.
- Fixed EOP exception condition caused by trying to retrieve `java LBMSourceStatistics`.
- Fixed condition that causes Windows to crash when debugging is enabled on 64-bit CPUs running 32-bit binaries.
- Fixed condition that prevented C++ example program from compiling.
- Fixed `umesrc -S` command line option to accept hostnames that are not fully qualified.
- Provided workaround that prevented `umestored` aborts by adding a check to the configuration file to not allow the asynchronous store write buffer length to be set to less than the largest fragment.
- Added check to `umestored` config at startup, to make sure that 90% of `repository-disk-file-size-limit` is greater than or equal to `repository-size-limit`, so that a fatal assert condition is avoided. If the check passes, then startup and normal operation continue. If it fails, then startup is aborted, the daemon will not start, and an error is logged.
- Fixed the `LBMWildcardReceiver.close` method (Java) so that it deletes the underlying native wildcard receiver.
- Fixed `SDM` condition in which adding a field with the same name as a deleted field fails. When deleting a field from a message, also delete the field name entry from the name lookup tree.

## 69. Release LBM 3.2 2007-08-27

- The **LBM** Gateway allows you to configure the use of blocking or non-blocking sends on the outbound direct side. The default is to use non-blocking sends. If a send blocks when using non-blocking sends, the message is dropped and the receiver is notified of loss; the notifying mechanism is the same one that notifies when a message exceeds the inbound Event Queue size or delay parameters.
- Avoid an array underflow that occurs when empty lines are encountered in configuration files.
- Fixed header parsing to prevent spurious messages, sent to an **LBM/UME** port, from forcing **LBM/UME** into an infinite loop.
- Fixed problem to prevent crash that occurs when debug logging is enabled on 64-bit Windows systems running 32-bit binaries.
- Added support for the `response_tcp_interface` context configuration attribute to specify TCP interface for outgoing response connections. Allow non-multicast capable interfaces (e.g., loopback) to be accepted by `request_tcp_interface`.
- Added the number of the failed port to bind-failure log messages.
- Fixed the condition where a receiver using arrival order delivery always drops the first message, if it is a fragment.
- Corrected condition that could cause a fatal assertion (`fdm->poll_fds!=NULL`) when `fd_management_type` is set to 'poll'.

## 70. Release LBM 3.2ea2 2007-07-19

- A number of internal changes to the **LBM** Gateway to better handle synchronization between the inbound and outbound sides, as well as reduce memory utilization.
- Changes to Windows completion ports to improve send performance. Also fixes potential heap corruption when receivers are deleted.
- Added config options `transport_lbtrm_source_socket_buffer` and `transport_lbtru_source_socket_buffer`. Set default values to 65536 or OS default, whichever is larger. This has the effect of improving Windows send performance.
- Added Windows support to the Gateway.
- Corrected the implementation of `-q` with Java and C#.
- Corrected the statistics omission from usage message on .NET (Ibm).
- Improved Completion Ports (send & receive).
- Added a `-s` (statistics) option to the Java version of `umercv`.
- Added Gateway to Solaris binary packaging.
- Fixed cleanup of callback token at JNI.
- Fixed the condition that crashed LBM, occurring when the same topic is allocated on separate threads and one thread creates and deletes its receiver before the second thread creates its receiver.



- Fixed condition where **LBM** allows corruption if more than 1024 FD's registered.
- Fixed condition where Gateway prints "topic mgmt packet invalid version" and ceases to function.
- Fixed condition where Gateway dumps core in mixed-receiver direct scenario.
- Fixed failed assertion [sock->rcv\_maxlen==maxlen] in Gateway.
- Fixed problem so that Gateway explicitly sets the context operational\_mode.
- Fixed a memory leak in the Gateway that occurred when large numbers of receivers were present.
- Fixed numerous problems in the Gateway that occurred when large numbers of receivers were present.
- Fixed an incorrect error message that could cause the Gateway to crash if debugging was on.
- Provided a fix that prevents spurious messages sent to an **LBM** port from throwing **LBM** into an infinite loop.

## 71. Release **LBM 3.2ea1 2007-06-15**

- Added Self-Describing Message C API.
- Added HyperTopic receiver C API.
- Added serialized response handle C API functions.
- Added Gateway support for Solaris.
- Added a debug flag to log all **LBM** API function calls.
- Added a configuration debug flag to the Gateway (lbgwd).
- Fixed the Java LBMSourceStatistics retrieval process so that it does not throw an EOP exception.

## 72. Release **LBM 3.1 / UME 1.1 2007-06-05**

- Fix Java LBMSourceStatistics which was broken in previous release.
- Fix Java LBMSourceStatistics/LBMReceiverStatistics.refresh() memory leak.
- Improve handling of initial unrecoverable loss to prevent CPU spiking of **UME** store on startup.
- Fix offset calculation when reading in cache to prevent cache corruption on 64-bit machines.
- Add StoreID to **LBM** channel protocol to allow source to recognize keepalives coming from the wrong store, thus masking a "dead" store with which the source is currently registered.
- Initialize **UME** store disk\_len parameter to avoid fatal assertion occurring when a retransmission is requested for a message stored on the disk.
- Ignore retransmission requests for unrecoverably lost messages.
- Force topic resolution traffic to be ignored until the **UME** store is ready to accept connections to avoid potential problems with inconsistent state resulting in a crash.

- Fix handling of the lead sequence number by no-cache **UME** store and for handling of sequence numbers by receivers for late join when a no-cache store is being used.
- Corrected a problem caused by fragmented (larger than 64K) response messages in which some internal fields of the message were not properly set, leading to a potential crash in certain circumstances.
- Added check of kernel version to **UME** stored on Linux, logging a warning if the kernel version is < 2.6.

## 73. Release LBM 3.1 / UME 1.1ea1 2007-04-20

- Fix **UME** store so that it requests the correct sequence number range from the source (store failover operation).
- Fix **UME** store daemon to be able to handle retransmission requests for messages retained on disk.
- Modify check for minimum size read when reading store cache files to work correctly on 64-bit OS to prevent segfault of **UME** store daemon and loss of persisted content.
- Modified setting of lead sequence number when reconnecting to **UME** store to avoid various bad behaviors, e.g., unchecked store memory growth, loss of persistence, unreclaimed source memory.
- Add check to prevent receiver from registering the same ID across multiple registered sources and change example **UME** receiver to assign its registration ID as an offset from the corresponding source ID.
- Reset timer id inside of LBMTimer callback so that subsequent calls to the `cancel()` method do not inadvertently cancel the wrong timer.
- Prevent wrong type of LBMTopic object from being passed to LBMSource, LBMReceiver, or LBMWildcardReceiver constructors.
- Prevent crash if `data()` method called for the first time outside of the C# LBMReceiver callback.
- Add support for ordered delivery and fragmentation/reassembly to MIM.
- Added serialization of response information.
- Added Self-Describing Message C API.

## 74. Release LBM 3.0 / UME 1.0 2007-02-16

- Added `ume_retransmit_request_maximum` configuration option
- Added `lbm_src_send_ex()` API function
- Added support for **LBM** Gateway Daemon (`lbgwd`) on Linux
- Modified deletion of unresolved wildcard receivers; deletion will not take place if a single-topic receiver exists with a topic matching the wildcard receiver's topic pattern.
- Added synchronization to LBT-RM and LBT-RU timer function to handle the end of the NAK ignore interval, to protect against transmission window updates.
- Added support for `LBM_SRC_EVENT_MESSAGE_RECLAIMED` source event.

- Update `lbm_context_retrieve_src_transport_stats_ex()` and `lbm_context_retrieve_rcv_transport_stats_ex()` to return a value in `*num` parameter if the original value was too small making these functions semantically compatible with `lbm_rcv_retrieve_all_transport_stats_ex()`.
- Fix problem with store daemon that would cause it to crash on 64-bit Linux after being shutdown and restarted.
- Add Hot Failover support to Java and .NET APIs.

## 75. Release LBM 3.0ea2 / UME 1.0ea2 2006-11-16

- Added `request_tcp_interface` configuration option
- Added option to enable setting of interface for individual **UME** stores in store daemon
- **UME** config option `ume_retention_unique_acknowledgements` renamed `ume_retention_unique_confirmations`
- Bug fix in **UME** store for recreation of receiver state from data in disk repositories
- Added a callback in **UME** sources invoked when forced reclamation of retained messages occurs
- Fixed assertion in **UME** store daemon thrown when a **UME** source using a constant registration ID restarts
- Bug fix in .NET API `LBMTimer cancel()`
- Bug fix for proper handling of immediate requests with long topic names on Solaris

## 76. Release LBM 3.0ea1 / UME 1.0ea1 2006-09-29

- Added **UME** source and receiver logic to **LBM** libraries.
- Added **UME** persistent store.
- Now allow zero for `resolver_active_threshold` which disables sources becoming inactive (forces indefinite advertisements).

## 77. Series 2.3 Releases

Key feature added in the 2.3 series releases:

- Statistics monitoring capability added to all APIs

## 77.1. Release 2.3.10 2007-01-17

- Fix to avoid unnecessary scheduling of the LBT-RU/RM rate controller once the retransmission rate limit has been reached.

## 77.2. Release 2.3.8 2006-12-21

- Added ABI support for future binary compatibility
- Bug fix for object deletion in JNI interface routines
- Bug fix in the setting of application IDs used in **LBM** statistics monitoring

## 77.3. Release 2.3.7 2006-12-14

- Added ABI support for future binary compatibility
- Bug fix for object deletion in JNI interface routines
- Bug fix in the setting of application IDs used in **LBM** statistics monitoring

## 77.4. Release 2.3.6 EA1 2006-12-01

- Bug fix for race condition in LBT-RU/LBT-RM rate controller

## 77.5. Release 2.3.5 2006-11-29

- Bug fix to prevent session messages from backing up the lead sequence number to a previous state
- Bug fix for proper handling of immediate requests with long topic names on Solaris
- Bug fix in `lbm_license_str()`; the input string is no longer modified in place.
- Added `dispose` method to `LBMMMessage` class in the Java API

## 77.6. Release 2.3.4 2006-10-30

- Added Event Queue attribute `queue_objects_purged_on_close` to control whether the Event Queue is "purged" when various object types (source, receivers, requests, wildcard receivers, timers) are closed
- Bug fix of off-by-one check for NAK buffer overflow

- .NET namespace changed to com.latencybusters.lbm
- .NET assembly (lbmcs.dll) signed with unique strong name
- Changed behavior of object-specific `getopt()` functions `lbm_src_getopt()`/`lbm_src_str_getopt()`, `lbm_rcv_getopt()`/`lbm_rcv_str_getopt()`, and `lbm_event_queue_getopt()`/`lbm_event_queue_str_getopt()` to return a value for all options regardless of whether they are settable via the corresponding `setopt` function
- Added **LBM** version preprocessor definitions

## 77.7. Release 2.3 2006-08-04

- Split `liblbmj.so` (`lbmj.dll`) from `liblbm.so` (`lbm.dll`)
- Added statistics monitoring capability to C API
- Added statistics monitoring capability to Java API
- Added statistics monitoring capability to .NET API
- Added **LBM** API functions to enable initialization of an **LBM** license from a string or file.
- Changed default value for `resolver_multicast_ttl` from 1 to 16
- Added a signature to verifiable messages (`-V` option) generated by example source programs, and checked for by example receiver programs. This prevents receivers, which are expecting verifiable messages, from incorrectly flagging every message from a non-verifiable message source as failing verification.

This change introduces an incompatibility (with regard to example programs using verifiable messages only) with previous releases. Example programs from earlier releases will not work with example programs from Release 2.3 (when using verifiable messages).

## 78. Series 2.2 Releases

Key features added in the 2.2 series releases:

- Microsoft .NET framework support providing APIs for all Visual Studio languages (including C#, J#, Visual Basic, and Visual C++).
- LBM Configuration Guide (`../Config/index.html`) added to documentation set.
- `lbmprice.c` added to examples (`../example/index.html`) section. A simulated price source/receiver application demonstrating new source notification, source/receiver awareness, arrival-order delivery, differential updates, and hot failover.

## 78.1. Release 2.2.2 2006-07-17

- Bug fix for the .NET API to handle null callback data argument for source events (specifically LBM.SRC\_LBM\_WAKEUP)

## 78.2. Release 2.2.1 2006-06-07

- Bug fix in lbmmrcv.java.
- Bug fix in LBT-RU on 64-bit machines.

## 78.3. Release 2.2 2006-05-23

- Key features listed above.

## 79. Series 2.1 Releases

Key features added in the 2.1 series releases:

- Hot Failover functionality (see API functions `lbm_hf_*()`).
- Completion Ports for Windows (see configuration option `context fd_management_type`).
- Aggregate Statistics (see API functions `lbm*_retrieve_all_transport_stats()`).
- Additional topic resolution configuration (see options `context resolver_maximum_advertisements`, `context resolver_cache` and `receiver resolution_number_of_sources_query_threshold`).
- Non-messaging contexts to support timers and I/O synchronization (see API function `lbm_context_reactor_only_create()`).
- New API functions to allow access to associated objects (see API functions `lbm_context_from_*()`, `lbm_event_queue_from_*()`, and `lbm_topic_from_*()`).
- A license key is now required for operation.

## 79.1. Release 2.1.1 2006-04-04

- Java performance improvements.

## 79.2. Release 2.1 2006-03-17

- Key features listed above.

## 80. Series 2.0 Releases

Key features added in the 2.0 series releases:

- Added wildcard support.

### 80.1. Release 2.0.6 EA1 2006-02-16

- Added support for 64-bit AIX

### 80.2. Release 2.0.5 EA2 2006-02-07

- Bug fix: thread ID was not being set when `lbm_context_process_events()` was called
- Bug fix: `lbmpong` reported incorrect round-trip times on 64-bit platforms due to a buffer overflow

### 80.3. Release 2.0.5 EA1 2006-01-30

- Added 64-bit Solaris-10 x86 platform.

### 80.4. Release 2.0.4 2005-01-09

- Fixed bug that prevented wildcard receivers from working properly with Event Queues. The symptom was:  
`FATAL: failed assertion [msg->refcnt>=0] ...`
- Added Event Queue option to Java `lbmwrcv` sample application
- Added `liblbmj.a` and extracted `liblbmj.so.0` for Java support on AIX. This requires J2RE 1.4.2 SR3 or later.

### 80.5. Release 2.0.3 2005-12-14

- Fixed Windows bug that occasionally caused the error:

FATAL: failed assertion [mapentry!=NULL] ...

## 80.6. Release 2.0.2 2005-12-06

- Modified interface scan to find aliased IP addresses on AIX

## 80.7. Release 2.0.1 EA1 2005-11-14

"Early Access" release.

- Added support for RHEL4 on Itanium/ia64

## 80.8. Release 2.0 2005-11-08

- Build java.util.Properties semantics into classes implementing configuration attributes
- Added **LBM** Start Menu Program Group to Windows installation
- Added `lbmspike` command
- Clarifications in `lbm-config.txt` (units given where appropriate)

## 80.9. Release 2.0 EA1 2005-09-30

"Early Access" release.

- Added wildcard support.
- Extract log4j and java.util.logging references from Java **LBM** class.
- Added support for Solaris 10/x86.
- Added callback that notifies the application when new topics/sources are discovered by topic resolution (see `resolver_source_notification_function` in `doc/Config/lbm-config.txt` for details).
- `lbmrd`: added "-i" option and removed "-H" option.
- Added `lbmwrcv` example for wildcard usage.
- Added QuickStart document.
- Added significant content to Design document.



## 81. Release 1.7 2005-08-23

- Add support for Java API using JNI layer
- Add Source Side Filtering support for TCP transport

## 82. Series 1.5 Releases

Key features added in the 1.5 series releases:

- Added Latency Busters Transport for Reliable Unicast (LBT-RU)
- Major improvements to example code and documentation
- Added support for AIX

### 82.1. Release 1.5.1.4 2005-07-11

- Minor revisions for AIX

### 82.2. Release 1.5.1.3 2005-07-08

- Fixed a bug that broke unicast topic resolution in release 1.5.1.
- Added support for AIX

### 82.3. Release 1.5.1 2005-06-03

- Added Latency Busters Transport for Reliable Unicast (LBT-RU)
- Major improvements to example code:
  - Added `-r` flag for easier testing with LBT-RM
  - Added `lbmresping` for troubleshooting **LBM** unicast resolver problems
  - Added Makefile (for Windows) and all code needed to compile examples.
  - General cleanup and many comments added
- Added five new sections to THPM white paper:
  - Messaging Latency Budget
  - Latency Sources

- Latency Measurement Overview
- Measuring CPU Scheduling Latency
- Measuring CPU Contention Latency
  
- Improvements and additions to the FAQ in the Application Notes
- Added example config files (see `doc/Config/Examples`) and moved `doc/lbm-config.txt` to `doc/Config/lbm-config.txt`.
- Added section to `lbm-config` that lists the ranges and port usage for TCP/UDP.
- Changed port ranges for some settings. Changes are:

Ports Used	Old	New
LBT-RM src unicast ports	4390 - 4300	4390 - 4399
Unicast resolver ports	4401 - 4405	4402 - 4406
LBT-RM destination port	4380	4400
TCP transport ports	4380 - 4390	4371 - 4390

- Added `request_tcp_reuseaddr` config variable.
- Fixed an error saving fragment state on platforms that require memory alignment (e.g. SPARC) that was causing assertion failures with loss.

## 83. Series 1.1 Releases

Key features added in the 1.1 series releases:

- Unicast topic resolution
- Message fragmentation and reassembly
- Immediate messaging
- Solaris support
- Documentation additions and improvements

### 83.1. Release 1.1.5.2 2005-05-23

- Removed check of 0 when setting TTL so that a TTL of 0 can be used for `resolver_multicast_ttl`.
- Fixed an error with creating new receivers when multiple sources are defined already by the topic. Was leading to assertion failure when cleaning up on receiver deletion.

## 83.2. Release 1.1.5 2005-04-05

- Added LBM\_EVENT\_QUEUE\_POLL arg to lbm\_event\_dispatch(). Also added LBM\_EVENT\_QUEUE\_ENQUEUE\_NOTIFICATION monitor function event type. Added queue\_enqueue\_notification config option to event queues.

## 83.3. Release 1.1.4.3 2005-03-21

- Changed default mim\_destination\_port to 4401 to avoid the default LBT-RM unicast source port range.
- Modified the LBT-RM multicast address range to be 224.10.10.10 to 224.10.10.14.
- Fixed forwarding of ads via lbmrd so that it fills in fields in the ads that are set to INADDR\_ANY for sources.

## 83.4. Release 1.1.4 2005-03-08

- Added unicast topic resolution support.
- Added ability to specify a custom string hash function to use for hash table manipulation for topics. Added the ability to set the hash table sizes.
- Added new options:
  - *resolver\_unicast\_address*
  - *resolver\_unicast\_interface*
  - *resolver\_unicast\_port*
  - *resolver\_unicast\_port\_high*
  - *resolver\_unicast\_port\_low*
  - *resolver\_unicast\_destination\_port*
  - *resolver\_unicast\_receiver\_socket\_buffer*
  - *resolver\_source\_map\_tablesz*
  - *resolver\_receiver\_map\_tablesz*
  - *resolver\_string\_hash\_function*

## 83.5. Release 1.1.3.1 2005-03-01

- Fixed byte ordering issue on destination port for multicast immediate messages.
- Added ChangeLog, TechPres, and GetStart to Doc package.

- Moved ReadMe from doc subdirectory to release directory. ReadMe is also now included in code packages in case they get separated from doc packages.

## 83.6. Release 1.1.3 2005-02-28

- Added fragmentation/reassembly to handle large messages for senders and responses (not for immediate messages).
- Added -V arg to lbmsrc and lbmrcv examples to verify message contents.

## 83.7. Release 1.1.2 2005-02-11

- Added Solaris support.
- Separate doc package from binary packages.
- Doc package contains AppNotes, **LBM** Design, THPM, examples, and ReadMe.
- Added *delivery\_control\_maximum\_burst\_loss*. Documented *delivery\_control\_loss\_tablesz* and *delivery\_control\_order\_tablesz*.
- Added `LBM_MSG_UNRECOVERABLE_LOSS_BURST` message type.

## 83.8. Release 1.1.1 2005-02-04

- Modified storage of topic name in topic structure so that it uses less memory for small topic names.
- Added warning to **LBM** API doc for functions that are not safe to call from context thread callbacks.
- Added -M I arg to lbmstrm example to have a sender that never stops sending data. Cleaned up lbmimsg and lbmireq to handle daemon mode operation.
- Added the usage of `ctime_r` and `localtime_r` when available. Fixed error in setting resolver multicast interface from the `setopt()` version.
- Changed default LBT-RM coalesce threshold to 15 to accommodate Solaris iov limitation. Also, on Solaris, when setting the value, it will not accept a value greater than 15.
- Added **LBM** daemon support to sending and receiving multicast immediate messages. Unicast immediate messages go directly to the target bypassing the daemon.

## 83.9. Release 1.1.0.1 2005-01-15

- All multicast immediate message config options added.
- Added basic immediate message support (unicast and multicast).

- Fixed bug with priorities when TCP sessions dequeue loaded queues.

## 84. Series 1.0 Releases

Key features in the 1.0 series releases:

- "No daemons" direct messaging design
- Multi-threaded design for high performance/low latency
- All message routing done in network hardware--not in software
- Basic messaging API; request/response API
- Managed Event Queue API; timer API
- Explicit/Implicit batching of messages
- Choice of transports: TCP, TCP-LB (latency-bounded TCP), or LBT-RM (latency-bounded reliable multicast).
- Automatic/Manual topic to transport mapping
- Multiple transports per sender
- Multicast topic resolution
- Linux/Windows platform support
- Configuration via API or config file

### 84.1. Release 1.0.3 2004-12-11

- Added latency bounded TCP behavior and *transport\_tcp\_multiple\_receiver\_behavior* option.

### 84.2. Release 1.0.2 2004-12-06

- Fixed stat retrieval on receiver so that it handled multiple sources correctly.
- Added Windows static library changes. The static **LBM** library is named `lbm_static.lib`.
- Added *response\_tcp\_nodelay* and *response\_tcp\_deletion\_timeout* options.

### 84.3. Release 1.0.1 2004-11-19

- Changed the pipe used in notifying `fdm` so that it is non-blocking on both ends. Helps to avoid a potential block on `write()` under some circumstances that can lead to deadlock.

- Added *transport\_lbtrm\_coalesce\_threshold* option to help with memory usage with small messages at the expense of memcpys.

## 84.4. Release 1.0 2004-10-29

- Updated LBM Design document.
- Added retransmit delay ignore information (including statistics) and added handling of the flag. LBT-RM now ignores after the packet has cleared the retransmit rate control mechanism instead of before.
- Added *transport\_tcp\_listen\_backlog* and *request\_tcp\_listen\_backlog* options.
- Added *transport\_lbtrm\_send\_naks* option.
- Added SIGPIPE ignore to a few other examples.
- Added len checks based on transport for sending. Changed default SO\_SNDBUF for TCP on Windows to be 128000 instead of OS default.
- Added LBM\_RESOLVER\_LOSS\_RATE environment variable. Added optional value for initial delay to lbmreq example. Modified lbmresp to take a -E arg to tell it to exit upon EOS (default is not to).
- Added retrieval of *request\_tcp\_port* and display of the value for lbmreq and lbmmreq examples.
- Added SIGPIPE ignore to lbmmsrc and added waiting for all threads to finish before deleting sources.
- Added changes to example receiver output display.
- Added stats retrieval functions and structures.