Ultra Messaging® Manager Guide

Copyright © 2011 - 2014 Informatica Corporation 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 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) Roque 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>. 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.

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/hsqlLicense.html,

http://httpunit.sourceforge.net/doc/ license.html, http://jung.sourceforge.net/license.txt ,

http://www.gzip.org/zlib/zlib_license.html, http://www.openIdap.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://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://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/EaseIJS/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; 7676516; 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.

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. Ultra Messaging Manager	4
3. Using the UMM GUI	6
4. Using the UMM API	
5. UM Manager Daemon	
6. Manpage for ummd	37

1. Introduction

The **Ultra Messaging**® **Manager** 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.

UM Manager employs a daemon (ummd) to serve license and configurations information to your applications. The **UM** Manager graphical interface (GUI) accesses the daemon via the UMM API. The **UM** Manager GUI also simplifies **UM** configuration by providing the flexibility to configure applications in highly similar or divergent ways.

Note: The UMM Daemon is not supported on the HP NonStop® platform.

2. Ultra Messaging Manager

Some of the highlights of the UM Manager include the following.

- Implements UM configuration with an XML file(s).
- · Provides a graphical interface (GUI) to create and edit XML configuration files.
- · Allows configuration by topic name, context name, event queue name, or wildcard pattern.
- Allows restriction of option values to a range or ranges of values.
- Allows you to restrict the creation of **UM** objects such as contexts, source topics, receiver topics, event queue names, or wildcard patterns.
- · Lets you define configuration templates, which can overlap and override one another.
- · Provides arbitrary strings of application-data at strategic points in the configuration.
- Retrieves configuration from a ummd, or via an API call, or via an environment variable (in that order).
- Preserves the existing UM configuration APIs (lbm_config(), *_attr_setopt(), etc).

Note: The **UM** Manager currently supports all **UM** features except the Hot Failover Across Multiple Contexts (../Design/lbm-features.html#HFX) (HFX) feature.

Note: The **UM** Manager does not provide policy enforcement for Spectrum channels (../Design/lbm-features.html#SPECTRUM). **UMM** rules apply to topic names, but not to Spectrum channels within a topic name. Moreover, you assign receivers to Spectrum channels with API calls and not **UM** configuration options.



Figure 1. Ultra Messaging Manager Architecture

UMM consists of the following components.

- UMM GUI Graphical user interface for creating and editing XML configuration files. The UMM GUI requires Java Version 1.6.x. Communicates with ummd via the UMM GUI API. See *Using the UMM GUI*.
- **UMM** GUI API Used to create the UM GUI and provides all access to the configuration database. You can use the API to programmatically create and store configuration templates and application configurations, bypassing the **UMM** GUI. See *Using the UMM API* and UMM Java API (../UMMJavaAPI/html/index.html).
- ummd Serves XML configuration information and UM license information to UM and Ultra Messaging JMS applications. Communication between ummd and your applications can be secured with SSL. See UM Manager Daemon.

 Configuration Database - Database that stores all XML configuration files created in the UMM GUI or created programmatically. UMM supports MySQL[™] and Oracle[®] databases using JDBC. See *Configuring the UMM Database*.

3. Using the UMM GUI

You can use the **UMM** GUI to create XML format configuration files if you do not want to create them manually with an XML editor. **UMM** stores any configuration information you create with the **UMM** GUI in the **UMM** configuration database. Normally, **UM** applications contact the UMM Daemon to obtain their configuration information. However, the GUI also offers a method to dump the XML data to a file which could be loaded by an **UM** application without involving the daemon.

See UMM Daemon (../QuickStart/start-daemons.html#START-UMMD) for information about starting the UMM daemon and GUI.

This section discusses the following topics.

- Using UMM With a UM Application
- Using UMM With a JMS Application
- UMM Menus
- Creating Configuration Templates
- Using the Order Attribute
- Creating Application Configurations
- Creating JMS Configurations
- Managing Users

Note: You can copy and paste a manually created UM XML configuration file in the XML View pane and then edit and store this configuration file using the UMM GUI.

3.1. Using UMM With a UM Application

The following procedure describes a general approach to using **UMM** to serve license and configuration values to your applications. This procedure contains general guidelines for configuring **UM** for the application SENDAPP.

1. Start the UMM Daemon and the **UMM** GUI. See UMM Daemon (../OuickStart/start-daemons.html#START-UMMD).

Note: The UMM Daemon verifies the license file specified in its XML configuration file and then serves it to applications that you configure with the **UMM** GUI. This provides a centralized method to manage your **UM** license. See *UM Manager Daemon*.

2. Create any desired templates to hold configuration option values shared by multiple applications or primitive **UM** objects (context, source, receiver, wildcard receiver or event queue). See *Creating Configuration Templates*.

You can create and apply multiple templates to applications and primitive **UM** objects, however, if the same option appears in multiple templates, the option value in the last template overrides the option value in the previous template. See *Applying Templates*.

- 3. Create an application for your **UM** application and apply any relevant templates created in the previous step. Just for this example, name the application, SENDAPP. See *Creating Application Configurations*.
- 4. For the SENDAPP application, configure the application's default context by applying any relevant templates and setting context option values.
- 5. Since our example application, SENDAPP is a sending application, also configure its Source options. See *Configuring Sources*. (If this was a receiving application, you would configure Receiver or Wildcard Receiver options.)
- 6. If your application creates multiple contexts, repeat the previous two steps for each context.
- 7. Configure the default Event Queue for SENDAPP, applying any relevant template and setting option values.
- 8. Assign user Admin to SENDAPP and give it the password, secret. See Managing Users.
- 9. Set the following environment variable on the machine where SENDAPP runs. See also *Connecting Your Applications to the UMM Daemon.*

Set LBM_UMM_INFO to SENDAPP:Admin:secret@10.29.3.95:21273.

Note: UMM Daemon runs on a machine at 10.29.3.95:21273.

10. Start SENDAPP.

3.2. Using UMM With a JMS Application

The following procedure describes a general approach to using **UMM** to serve license and configuration values to a JMS application. This procedure contains general guidelines for configuring **UM** for the application UM_PRODUCER.

1. Start the UMM Daemon and the UMM GUI. See UMM Daemon (../QuickStart/start-daemons.html#START-UMMD).

Note: The UMM Daemon verifies the license file specified in its XML configuration file and then serves it to applications that you configure with the **UMM** GUI. This provides a centralized method to manage your **UM** license. See *UM Manager Daemon*.

2. Create any desired templates to hold configuration option values shared by multiple ConnectionFactories. For example you could create separate templates for using LBTRM or LBTRU as transports. See *Creating Configuration Templates*.

You can create and apply multiple templates to applications and primitive **UM** objects, however, if the same option appears in multiple templates, the option value in the last template overrides the option value in the previous template. See *Applying Templates*.

- 3. Create a ConnectionFactory and apply any relevant templates created in the previous step. Just for this example, name the ConnectionFactory, uJMSConnectionFactory. See *Configuring ConnectionFactories*.
- 4. For the uJMSConnectionFactory ConnectionFactory, configure options that are common to all the producers and consumers created by the applications using this ConnectionFactory. This would include options for contexts, sources, receivers and wildcard receivers.
- 5. Configure the Factory options for uJMSConnectionFactory. See Configuring Factories.
- 6. If uJMSConnectionFactory supports Destinations of:
 - Type=UMS, configure the appropriate options in the UMS tab, such as transports and store configuration options.
 - Type=UMP, configure the appropriate options in the UMP tab, such as store configuration options.
 - Type=UMQ, configure the appropriate queue options in the UMQ tab.
- 7. Create and configure the Destinations to be used by the producers and consumers created by applications using this ConnectionFactory. See *Configuring Destinations*.
- 8. Create a JMS user and assign uJMSConnectionFactory to the new user. Assign the password, jmssecret.
- 9. In the jndi.properties file, ensure that you specify the address:port where the UMM Daemon runs. For example,

ummd = 10.29.3.95:21273.

Note: UMM Daemon runs on a machine at 10.29.3.95:21273. See also *Connecting Your Applications to the UMM Daemon*.

10. Start JMS application(s).

3.3. UMM Menus

The following table explains the UMM menu selections.

Menu	Selection	Description
File	New	Allows you to create a new Application, Template or User.
File	Save	Saves the current Application, Template or User record.
File	Save All	Saves all the records that you have created or edited.

Menu	Selection	Description
File	Revert	Revert to the previously saved version of the Application, Template or User.
File	Save To File	Save the Application, Template or User record to an XML file instead of the database. This selection allows you to use the UMM GUI to configure applications but not use UMM Daemon when running your applications. After saving an application's configuration to a file, you can use the LBM_XML_CONFIG_FILENAME and LBM_XML_CONFIG_APPNAME environment variables to enable the XML configuration file. See Using XML Configuration Files With a UM Application (/Config/config.html#GETTINGSTARTEDXMLCONFIG).
File	Exit	Close the UMM GUI.
Edit	Duplicate	Copy the selected Application, Template or User record to a new record.
Edit	Delete	Delete the selected Application, Template or User record(s).

3.4. Creating Configuration Templates

You can define option templates that can be applied in application configurations. Multiple templates can be applied to any primitive UM object or application. Templates can be applied to most tags including <application>, <contexts>, <contexts>,

- 1. Click on Templates in the object tree in the left pane and then right-click or click on the File Menu.
- 2. Select New and then select Templates from the drop-down menu.
- 3. Enter the Template Name and click OK. The Template window appears in the right pane.

Figure 2. New Template Window

🛄 Ultra Messaging Manager					- 0 ×
File Edit Help					
rot rot High_Throughput Low_Latency Levs_Latency Applications Just Just Just Just Just Just Just Just AuditLog	Context Source Receiver EventQueue WildCard Receiver	Name	Value	Ordering	Permissions
				Options	
	Graphical Viev	W XML View			

4. Click on the Context tab and then click on Options button. A list of Context scope options appears.

Figure 3. Options List



- 5. Click on the checkbox next to the desired context options. Click **OK**. The selected options appear in the Template/Context pane.
- 6. For each option, click on the Value field, select or enter a new value and press ENTER.

😡 Ultra Messaging Manager	u Ultra Messaging Manager						
File Edit Help							
root	Context		Name	Value	Ordering		Permissio
High Throughput	Source	×	fd_management_type	wsaeventselect		Edit	
Low_Latency EventQueue	Receiver	×	resolver_multicast_address	224.9.10.11		Edit	
SendingLBTRM	WildCard						
∽ ☐ Users	Receiver						
⊶ 🗂 JMS							
- 🖸 Audit Log							
		-		Ontions	1		
				Options			
• • •	Graphical Viev	N	XML View				

Figure 4. Setting Option Values

- 7. If desired, set permissions for the option's values. See Setting Permitted and Restricted Values.
- 8. Repeat Steps 4-7 for Source Options.
- 9. Repeat Steps 4-7 for Receiver Options.
- 10. Repeat Steps 4-7 for Event Queue Options.
- 11. Repeat Steps 4-7 for Wildcard Receiver Options.
- 12. Click XML View to verify your entries.
- 13. Right click on the object name in the left pane or click on the File Menu. Then click Save.

3.5. Using the Order Attribute

Any option used in a template or directly configured for an application has an order attribute expressed with two values in one of two sequences,

```
<option default-value="tcp" name="transport" order="deny,allow">
```

or

```
<option default-value="tcp" name="transport" order="allow,deny">
```

The order attribute allows you to permit or restrict more than one value for the option. The order, **allow,deny** essentially means "allow what you specify, deny everything else." The order, **deny, allow** means "deny what you specify, allow everything else." The default order is **deny,allow**. If you specify nothing, the default rule executes, ignoring the option's default value. Order of appearance of rules is not significant.

3.5.1. Using the Deny, Allow Order

The example below allows any transport type except for LBT-IPC or LBT-RDMA.

```
<option default-value="tcp" name="transport" order="deny,allow">
    <deny>LBT-IPC</deny>
    <deny>LBT-RDMA</deny>
</option>
```

The procedure below shows how to accomplish this in the **UMM** GUI. Refer to the composite screen image following the procedure.

- 1. For the source option, transport, click in the Ordering box and select Deny, Allow.
- 2. Click the Edit button. The Permissions Window appears.
- 3. Click Add and type LBT-IPC.
- 4. Click again on Add and type LBT-RDMA.
- 5. Click OK.

Figure 5. Ordering - Deny, Allow

stating and sought	aller fallen i	100 ¹⁰			x
Name	Value	Ordering		Permissions	
implicit_batching_minimum_leng	h 2048		Edit		
× late_join	0		Edit		
× transport	tcp	Deny,Allow	Edit	[deny LBT-IPC, deny LBT-RDMA]	
X mplot_batching_minimum_length 2048 Edit X tate_join 0 Edit X tansport tzp DenyAllow Edit V DenyAllow Edit Edit V DenyAllow Edit Edit V DenyAllow Edit Edit V Deny LBT-PC Value V Deny LBT-PCC V Deny LBT-PCA V Deny LBT-RDUA V Delete Delete All OK Cancel V					

3.5.2. Using the Allow, Deny Order

You could accomplish the same restriction shown in Using the Deny, Allow Order with the next example.

```
<option default-value="tcp" name="transport" order="allow,deny">
  <allow>TCP</allow>
  <allow>TCP-LB</allow>
  <allow>LBT-RU</allow>
  <allow>LBT-RM</allow>
  </option>
```

The procedure below shows how to accomplish this in the **UMM** GUI. Refer to the composite screen image following the procedure.

- 1. For the source option, transport, click in the **Ordering** box and select **Allow,Deny**.
- 2. Click the Edit button. The Permissions Window appears.
- 3. Click Add and type TCP.
- 4. Click again on Add and type TCP-LB.
- 5. Click Add and type LBT-RU.
- 6. Click again on Add and type LBT-RM.
- 7. Click OK.



	agent, conserver, shifting the	televing too and or	ict disacting		
_					
	Name	Value	Ordering		Permissions
×	implicit_batching_minimum_length	2048		Edit	
×	late_join	0		Edit	
×	transport	tcp	Allow,Deny	Edit	[allow TCP, allow TCP-LB, allow LBT-RU, allow LBT-RM]
	Options Templates	Allow V LB	P P-LB T-RU T-RM 1 Delete OK Ca	Value Delete	

3.5.3. Setting Permitted and Restricted Values

- 1. To use the **Order** attribute for an option, click on the **Order** field and select either **Allow,Deny** or **Deny,Allow** (default).
- 2. Click on the Edit button in the Permissions column. The Permissions dialog box appears.
- 3. To add an option value with a specific permission, click Add. A permission row appears.
- 4. Select the permission from the Allow, Deny column.
- 5. Enter the option value in the Value column.
- 6. Repeat Steps 3-5 for each additional option value that requires specific permissions.
- 7. Click **OK**. The Permissions dialog box disappears and your entries appear in the Permission column for the option.
- 8. Click XML View to verify your entries.
- 9. Right click on the object name in the left pane or click on the File Menu. Then click Save.

3.5.4. Options with Value Ranges

Some options can specify a range of values. You can specify both allow and deny rules to allow one or more values of a larger range. The next example allows a single multicast address out of a range of otherwise allowable addresses.

```
<options>
  <option name="transport_lbtrm_multicast_address" default_value="239.191.10.11" order="deny,allow">
        <allow>239.191.10.15</allow>
        <deny>239.191.10.12-239.191.10.20</deny>
        </option>
   </option>
```

Address 239.191.10.15 matches both an allow and deny rule, but since the order indicates that allow is the default, the value is allowed.

3.6. Creating Application Configurations

You can configure UM Configuration Options for specific applications. You can attach a template or select any number of individual options. Separate applications can have the same options, but with different values. You can also configure an application with multiple contexts or event queues and apply an order attribute to easily vary similar configurations.

- 1. Click on Applications in the object tree in the left pane and then right-click or click on the File Menu.
- 2. Select New and then select Application from the drop-down menu.
- 3. Enter the Application Name and click OK. The initial, blank Application window appears.

😡 Ultra Messaging Manager			- 0 X
File Edit Help			
root	Default Cont	text Event Queue Application	
P Applications Cf1 D D1	Context Sources	Name Value Ordering	Permissio
NewApp Stan	WildCard Receivers		
 			
- 🔝 Audit Log			•
		CPermission Comment	
			•
		Ordering For All Contexts	
		Deny,Allow V	•
	Graphical Vie	ew XML View	

Figure 7. New Application Window

4. Click on the Application tab. The Application window appears.

Figure 8. Application Window

Ultra Messaging Manager		
File Edit Help		
root	Default Context Event Queue Application	
Applications	Contexts	Event Queues
- 🗂 Users - 🗂 JMS Radit Log		
	Add Delete Edit	Add Delete Edit
	Templates	
	Comment	
	Graphical View XML View	

- 5. Add the additional contexts and event queues required by this application.
 - Click on the Add button and enter the context's or event queue's name.
 - Click OK.
 - Repeat the above steps for every additional context and event queue.
- 6. If you wish to apply a template to the application, click on the **Templates** button. The Templates window appears.

M Templates				×
High_Throughput		⇒		
Low_Latency Sending1 BTRM		6		
CondingLoritan		-		
	-	U		
		Û		
			-	
	OK	Cancel		

Figure 9. Templates List

See Applying Templates.

- 7. Click on the **Comment** text window and enter any notes or comments about the application. (These comments appear within the <application-data> element.)
- 8. Click on the **Default Context** tab.

🔝 Ultra Messaging Manager						- 0 X
File Edit Help						
File Edit Help Trot ← Templates ♥ Applications ♥ D 1 ► D1 ► D1 ► D1 ► D1 ► D1 ► D2 ► D1 ► D1 ► D2 ► D1 ► D2 ► D1 ► D2 ► D2	Default Context Context Sources Receivers WildCard Receivers	ext Event Queue App Na (Options Ten Permission	aplates	Value	Ordering	Permissio
		Allow				<u> </u>
		Ordering For All Contexts	3			
		Deny,Allow 💌	Comment			
4	Graphical Vie	W XML View				

Figure 10. Default Context Window

9. If needed, change the default *Ordering for All Contexts* to **allow,deny**. See also Using the Order and Rule Attributes (../Config/xmlconfigurationfiles.html#ORDER-RULE-ATTR).

10. Configure the application's Default Context.

- If you wish to apply a template to the Default Context, click on the **Templates** button and apply the template(s).
- Configure the Sources. See Configuring Sources.
- Configure the Receivers. See Configuring Receivers.
- Configure any Wildcard Receivers. See Configuring Wildcard Receivers.
- Configure the options for the Default Context by clicking on the **Options** button and selecting the options. See *Creating Configuration Templates* for more on configuring options.
- Set the **Rule** attribute, if needed.
- 11. Repeat the above step for all contexts in the application.
- 12. Click on the Event Queue tab. The Event Queue window appears.

Figure 11. Setting Option Values

😡 Ultra Messaging Manager		3
File Edit Help		
root	Default Context Event Queue Application	
Applications Applications Applications D1 D1 Stan Users AuditLog	Name Value Ordering Permissions	
	Options Templates	
	Permission	
	Allow	
	Ordering For All Event-queues	
	Deny Allow	
(Graphical View XML View	_

- 13. If needed, change the default *Ordering for All Event-queues* to **allow,deny**. See also Using the Order and Rule Attributes (../Config/xmlconfigurationfiles.html#ORDER-RULE-ATTR).
- 14. Configure the application's Default Event Queue.
 - If you wish to apply a template to the Default Event Queue, click on the **Templates** button and apply the template(s). queue's name.
 - Configure the options for the Default Event Queue by clicking on the **Options** button and selecting the options. See *Creating Configuration Templates* for more on configuring options.
 - Set the **Rule** attribute, if needed.
- 15. Repeat the above step for all Event Queues used in the application.
- 16. Right click on the object name in the left pane or click on the File Menu. Then click Save All.
- 17. Click on the **XML View** tab to review the actual XML configuration for the application. (Template options and values do not appear in this view.)

3.6.1. Configuring Sources

You can configure an individual UM source using a topicname or a topic pattern. Configuring a pattern allows you to easily specify option values for all the topics that match the pattern instead of configuring individual topics (topicname) with the same options and values. (You could also apply a template to the context that has all the source scope option values.) When using a topic pattern, the following rules apply.

- Topics that match both a topicname and a topic pattern use the option values from the more specific match, i.e. the topicname.
- If multiple patterns match a given topic, **UMM** uses the first pattern matched in the XML configuration file. In the example below, when your application creates a source for topic SRM, it will use the LBTRM transport, which is configured for the the first pattern, ***R***, and not the TCP transport which is configured for the second pattern, ***RM**.

```
<sources order="deny,allow">
  <topic pattern="*R*" rule="allow">
    <options type="source">
        <option default-value="lbtrm" name="transport"/>
        </options>
  </topic>
  <topic pattern="*RM" rule="allow">
        <option stype="source">
        <option default-value="transport"/>
        <option default-value="transport"/>
        <option default-value="transport"/>
        </options>
   </topic>
  </topic>
```

Use the following procedure to configure options for an individual UM source.

1. Click on the Sources tab on the left of the context pane. An empty Sources window appears.

File Edit Help Tool Octient Application Application Application Context Topic Sources Receivers Application Audit Log Add Default Context Cont	🛄 Ultra Messaging Manager							
□ tot ○ Tempidats ○ Tempidats ○ Oriext ○ Diant Context Topic ○ Diant Context Topic ○ Diant Context Receivers ○ ban Receivers ○ Diant Context Receivers ○ Audt Context Edit ○ Ordering For All Sources Comment □ Dent Allow ○	File Edit Help	File Edit Help						
Add X Delete Edit Ordering For All Sources Comment Deny Allow	File Edit Help root oct Applications 0 <tr< th=""><th>Default Context Event Queue Application Context Topic Sources Receivers WidGard Receivers</th></tr<>	Default Context Event Queue Application Context Topic Sources Receivers WidGard Receivers						
		Add Delete Comment Deny Allow T						

Figure 12. Sources Tab

- 2. If needed, change the default *Ordering for All Sources* to **allow,deny**. See also Using the Order and Rule Attributes (../Config/xmlconfigurationfiles.html#ORDER-RULE-ATTR).
- 3. Click on the Add button. The Source window appears.

Name	
Туре	
	Topic Name
Options	
implicit b	atching interval
implicit b	atching minimum length
implicit b	atching type
late join	3_1/20
resolver	advertisement maximum initial interval
resolver	advertisement minimum initial duration
resolver	advertisement minimum initial interval
resolver	advertisement minimum sustain duration
resolver	advertisement send immedidate response
resolver	advertisement sustain interval
resolver_	send_initial_advertisement
retransmi	t_retention_age_threshold
🔲 retransmi	t_retention_size_limit
🔲 retransmi	t_retention_size_threshold
Transport	
Transport	_lbtipc_behavior
transport	_lbtipc_id
transport	_lbtipc_maximum_receivers_per_transport
transport	Ibtipc sm interval
	Soloct All Cloar All
	Select All Cledi All

Figure 13. Source Topic Name/Pattern Options List

- 4. Enter the topicname or a topic pattern.
- 5. Select the **Type**, *Topic Name* or *Pattern* from the drop-down menu.
- 6. Click on the checkbox next to the desired options for the topic or pattern. Click **OK**. The **Sources** window appears with the selected options.

) Ultra Messaging Manager							
File Edit Help							
root	Default Cont	Default Context Event Queue Application Context Topic Name Value Ordering					
- C Applications	Context						
- 🕒 d1	Sources	ES ICXM implicit_batching_minimum_length 2048					
- 🕒 D1	Receivers		×	late_join	0		
- D stan	Receivers		×	transport	tcp		
		Add R Delete Sources Comment Deny Allow	-P(Diptions Templates		A V	
• • •	Graphical Vie	w XML View					

Figure 14. Topic/Pattern Option Values

- 7. Adjust the option values. See Creating Configuration Templates for more on configuring options.
- 8. If you wish to apply a template to the source, click on the **Templates** button and apply the template(s).
- 9. Set the **Rule** attribute, if needed.
- 10. Repeat Steps 3-9 for all topics and topic patterns for the context.

3.6.2. Configuring Receivers

Use the following procedure to configure options for an individual UM receiver.

1. Click on the Receivers tab on the left of the context pane. An empty Receivers window appears.

🔝 Ultra Messaging Manager	
File Edit Help	
Paie Solit Help Foot or Templates ← Applications ⊕ off ⊕ off ⊕ bits ⊕ bits ⊕ uMSConnectiont ⊕ UMSS ⊕ JMS ⊕ Addit Log	Default Context Event Queue Application Context Topic Sources WidGard Receivers
4 H	Crophra Waw YHI View

Figure 15. Receivers Tab

- 2. If needed, change the default *Ordering for All Receivers* to **allow,deny**. See also Using the Order and Rule Attributes (../Config/xmlconfigurationfiles.html#ORDER-RULE-ATTR).
- 3. Click on the Add button. The Receiver window appears.

Type Topic N	ame 💌			
Type Topic N Options	ame 💌			
Topic N	ame 🔻			
Options				
channel_map_tablesz	-			
delivery_control_loss_che	ck_interval			
delivery_control_maximum	_burst_loss			
hf_duplicate_delivery	-			
hf_optional_messages				
null_channel_behavior	_			
ordered_delivery				
resolution_no_source_notification_threshold				
resolution_number_of_sou	rces_query_threshold			
resolver_query_maximum	_initial_interval			
resolver_query_minimum_	initial_duration			
resolver_query_minimum_	initial_interval			
resolver_query_minimum_	sustain_duration			
resolver_query_sustain_in	terval			
retransmit_initial_sequenc	e_number_request			
retransmit_message_cach	ning_proximity			
retransmit_request_gener	ation_interval			
retransmit_request_interval	al _			
retransmit request maxin	num			
Select All	Clear All			
OK	Cancel			

Figure 16. Receiver Topic Name/Pattern Options List

- 4. Enter the topicname or a topic pattern.
- 5. Select the Type, Topic Name or Pattern from the drop-down menu.

6. Click on the checkbox next to the desired options for the topic or pattern. Click **OK**. The **Receivers** window appears with the selected options.

🛄 Ultra Messaging Manager					_		
File Edit Help	File Edit Help						
root	Default Cont	ext Event Queue Application					
Or Department Operations	Context	Торіс		Name	Value	Order	
- D d1	Sources	IXXB	×	ordered_delivery	1		
- 🕒 D1	Receivers		×	retransmit_request_interval	500		
- D stan	Receivers		×	retransmit_request_maximum	0		
UJMSConnection			×	retransmit_request_outstanding_maximum	200		
⊶ ⊡ Users ∽ ⊡ JMS							
Audit Log					•		
				Options Templates			
			P	Comment			
		🕂 Add 🛛 🗙 Delete 🧳 Edit		Allow		Î.	
		Ordering For All Receivers					
		Comment					
		Deny,Allow 🔻					
						······	
	Graphical View XML View						

Figure 17. Topic/Pattern Option Values

- 7. Adjust the option values. See Creating Configuration Templates for more on configuring options.
- 8. If you wish to apply a template to the source, click on the Templates button and apply the template(s).
- 9. Set the **Rule** attribute, if needed.
- 10. Repeat Steps 3-9 for all receiver topics and topic patterns for the context.

3.6.3. Configuring Wildcard Receivers

To apply XML configuration to a wildcard receiver, be sure both the pattern and pattern-type match exactly. You may also specify both receiver and wildcard receiver options.

Use the following procedure to configure options for an individual UM wildcard receiver.

1. Click on the **Wildcard Receivers** tab on the left of the context pane. An empty **Wildcard Receivers** window appears.

Figure 18. Sources Tab

📖 Ultra Messaging Manager	-	
File Edit Help		
root	Default Cont	ext Event Queue Application
Applications	Context	Pattern
- 🕒 cf1	Sources	
- 🕒 D1	Receivers	
NewApp	WildCard	
UJMSConnection	Receivera	
🕶 🗂 Users		
⊶ 📑 JMS		
- Lo Audit Log		
		+ Add 🗵 Delete 🥒 Edit
		Ordering For All Wildcard-receivers
		Comment
		Deny,Allow 🔻
	Graphical Vie	w XMI View
	anapineur en	

- 2. If needed, change the default *Ordering for All Wildcard-receivers* to **allow,deny**. See also Using the Order and Rule Attributes (../Config/xmlconfigurationfiles.html#ORDER-RULE-ATTR).
- 3. Click on the Add button. The Wildcard Receivers window appears.

Figure 19.	. Wildcard	Receiver	Pattern	Options	List
------------	------------	----------	---------	---------	------

Pattern Type pcre pcre Options pattern Type presolver, no.source_linger_timeout resolver, no.source_linger_timeout resolver, query_miniumum_interval channel_map_tablesz delivery_control_oss_check_interval delivery_control_maximum_interval indelivery_control_maximum_interval indelivery_control_maximum_interval iresolvtoin_no_source_notification_threshold resolvtoin_motery_miniumum_inital_interval resolver_query_minital_interval resolver_query_minital_interval resolver_query_minital_interval resolver_query_minital_interval resolver_query_minital_interval Select All Clear All	Pattern					
Pattern Type Porte Porte Options Options Postern_type resolver_no_source_linger_timeout resolver_no_source_linger_timeout resolver_query_minitum_interval resolver_query_minitum_interval delivery_control_noss_check_interval delivery_control_maximum_stal_sous ht_duplicate_delivery ht_optional_behazions th_duplicate_delivery resolution_no_source_notification_threshold resolver_query_minitum_initial_interval resolver_query_minitum_initial_duration resolver_query_minitum_initial_interval resolver_mery_minitum_initial_interval resolver_mery_minitum_initial_interval resolver_query_minitum_istial_duration resolver_query_minitum_istial_interval Select All Clear All	E					
	Pattern Type					
Options partial state of the second state of	pcre 💌					
stern_type resolver_no_source_finger_timeout resolver_opc.source_finger_timeout resolver_opery_maintum_interval resolver_opery_minitum_interval delivery_control_toss_check_interval delivery_control_maximum_purst_loss h_d_optional_messages null_channel_behavior ordered_delivery resolution_no_source_notification_threshold resolver_opery_minitum_inital_interval resolver_opery_minitum_inital_iduration resolver_opery_minitam_inital_iduration resolver_opery_minitam_inital_iduration resolver_opery_minitam_inital_iduration resolver_opery_minitam_inital_iduration reso	Options					
resolver_no_source_ilinger_timeout resolver_query_mainmum_interval resolver_query_minimum_interval channet_map_tablesz delivery_control_loss_check_interval delivery_control_loss_check_interval delivery_control_loss_check_interval delivery_control_loss_check_interval delivery_control_loss_check_interval delivery_control_loss_check_interval ordered_delivery mit_channet_behavior ordered_delivery resolution_no_source_notification_threshold resolver_query_minimum_inital_interval resolver_query_minimum_inital_duration resolver_query_minimum_istal_iduration resolver_query_minimum_istal_iduration resolver_query_minimum_istal_iduration resolver_query_minimum_istal_iduration resolver_query_minimum_istal_iduration resolver_query_minimum_istal_iduration resolver_query_minimum_istal_iduration resolver_query_minimum_istal_iduration resolver_query_minimum_istal_iduration	pattern_typ	e	-			
I'resolver_query_maintumu_interval I'resolver_query_minitumu_interval I'resolver_query_minitumu_interval I channel_map_tablesz I delivery_control_oss_check_interval I delivery_control_maximum_jurst_loss I hf_optional_messages I nil_otablesz I ordered_delivery I nil_otablesz I ordered_delivery I nil_otablesz I resolvion_no_source_notification_threshold I resolvion_query_minitum_inital_interval I resolver_query_minitum_statia_duration I resolver_query_minitum_statia_iduration I resolver_query_minitumu_statia_iduration I resolver_query_minitumu_statia_iduration I resolver_query_minitumu_statia_iduration I resolver_query_minitumu_statia_iduration I resolver_duery_minitumu_statia_iduration I resolver_duery_mi	resolver_no	_source_linger_timeout				
resolver_query_minimum_duration resolver_query_minimum_literval channel_map_tablesz delivery_control_loss_check_interval delivery_control_loss_check_interval delivery_control_loss_check_interval delivery_control_loss_check_interval ordered_delivery ordered_delivery ordered_delivery resolution_ne_sources_query_hreshold resolver_query_minimum_inital_interval resolver_query_minimum_inital_duration resolver_query_minimum_inital_interval resolver_query_minimum_inital_interval resolver_query_minimum_inital_interval Select All	🔲 resolver_qu	ery_maximum_interval				
resolver_query_minimum_interval delivery_control_Joss_check_interval delivery_control_Joss_check_interval delivery_control_maximum_bert_Joss hf_dplicate_delivery resolution_no.source_notification_threshold resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval	🔲 resolver_qu	ery_minimum_duration	1			
channel_mop_tablesz delivery_control_loss_check_interval delivery_control_loss_check_interval ht_duplicate_delivery Int_optional_messages ordered_delivery resolution_number_of_sources_query_threshold resolver_query_innaimm_mitial_interval resolver_query_innimm_mitial_interval resolver_query_innimm_mitial_interval resolver_query_innimm_mitial_interval resolver_query_innimm_mitial_interval resolver_query_innimm_mitial_interval resolver_query_innimm_mitial_interval resolver_query_innimmum_interval resolver_query_innimmum_interval resolver_query_innimmum_interval resolver_query_innimmum_interval resolver_query_innimmum_interval resolver_query_innimmum_interval resolver_query_innimmum_interval	🔲 resolver_qu	ery_minimum_interval				
delivery_control_pass_check_interval delivery_control_maximum_burst_joss hf_dplicate_delivery hf_optional_messages null_channel_behavior ordered_delivery resolution_no_source_ontflication_threshold resolution_number_of_sources_query_threshold resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval	Channel_ma	ap_tablesz				
delivery_control_maximum_burst_loss ht_duplicate_eldevry ht_optional_messages ordered_delivery resolution_number_of_sources_query_threshold resolution_number_of_sources_query_threshold resolver_query_minimum_minital_ideratal resolver_query_minimum_substal_ideratal resolver_query_minimum_substal_ideratal resolver_query_minimum_substal_ideratal resolver_query_minimum_substal_ideratal resolver_query_minimum_substal_ideratal resolver_query_minimum_substal_ideratal resolver_query_minimum_substal_ideratal resolver_query_minimum_substal_ideratal resolver_query_minimum_substal_ideratal	delivery_control_loss_check_interval					
Mr. dpilotate_delivery Mr. optionat_messages null_channel_behavior ordereid_delivery resolution_no_source_ondress_query_threshold resolver_query_maximum_initial_interval resolver_query_minimum_initial_interval	delivery_co	ntrol_maximum_burst_loss				
hr.optional_messages indicates behavior ordered_delivery resolution_oscurce_notification_threshold resolution_no_scurces_notification_threshold resolver_query_mainum_initial_interval resolver_query_mainum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver query_minimum_initial_interval resolver query_minimum_interval resolver query_minimum_initial_interval resolver query_minimum_initial_interval resolver query_minimum_interval resolver query_minimum_interval resolver	hf_duplicate	e_delivery				
null_channel_behavor ordered_delivery_notification_threshold resolution_no_source_guery_threshold resolution_umber_of_sources_guery_threshold resolver_query_minimum_inital_interval resolver_query_minimum_inital_interval resolver_query_minimum_inital_interval resolver_query_minimum_inital_interval resolver_query_minimum_inital_interval resolver_query_minimum_inital_interval resolver_query_minimum_inital_interval	m_optional_messages					
ordered_deavery resolution_source_notification_threshold resolution_number_of_sources_query_threshold resolver_query_mainmum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval	null_channel_behavior					
resolution_no_Source_nomication_intreshold resolution_mumber of_sources_gency_threshold resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_usitial_interval resolver_query_minimum_usitial_interval resolver_query_minimum_usitial_interval	ordered_de	livery				
esonume_inter_inte	resolution_	no_source_notification_threshold				
resolver_queer_nninnum_mitial_duration resolver_queer_nninnum_mitial_duration resolver_queer_nninnum_mitial_interval resolver_queer_sustain interval resolver_queer_sustain interval Select All Clear All		number_or_sources_query_uneshold				
resolver_query_minimum_initial_interval resolver_query_minimum_initial_interval resolver_query_minimum_sustain_duration resolver_query_sustain_interval Select All Clear All	resolver_qu	vory minimum initial duration				
resolver query minimum sustain_duration resolver query sustain interval Select All Clear All	resolver_qu	ery minimum initial interval				
esolver query sustain interval	resolver_qu	ery minimum sustain duration				
Select All Clear All	resolver qu	ery sustain interval	-			
Select All Clear All						
		Select All Clear All				
OK Cancel		OK Cancel				

- 4. Enter the wildcard receiver **Pattern**.
- 5. Select the **Pattern Type**, *pcre*, *regex* or *appcb* from the drop-down menu. See <code>pattern_type</code> (../Config/wildcardreceiveroptions.html#WILDCARDRECEIVERPATTERNTYPE) for more information about these option values.
- 6. Click on the checkbox next to the desired options for the topic or pattern. Click **OK**. The **Wildcard Receivers** window appears with the selected options.

Ultra Messaging Manager						• X
File Edit Help						
root	Default Cont	ext Event Queue Application				
- I emplates	Context	Pattern		Name	Value	Order
- D cf1	Sources	PM	×	pattern type	appcb	
- 🕒 D1	Receivers		5	ordered delivery	1	
— 💭 NewApp	WildCard		8	ordered_delivery	1000	
- 🕒 stan	Receivers			resolver_query_sustain_interval	1000	
			×	retransmit_request_interval	500	
er 🛄 JMS Ruditog	Add S Delete Ordering For All Wildcard-receivers Comment Deny,Allow	Add X Delete Fdit Ordering For All Wildcard-receivers Comment Deny,Allow	• -Pe	Dptions Templates		
	Graphical Vie	w XML View				

Figure 20. Wildcard Receiver Pattern Option Values

- 7. Adjust the option values. See Creating Configuration Templates for more on configuring options.
- 8. If you wish to apply a template to the source, click on the Templates button and apply the template(s).
- 9. Set the **Rule** attribute, if needed.
- 10. Repeat Steps 3-9 for all wildcard receiver patterns for the context.

3.6.4. Applying Templates

The Templates dialog box allows you to apply templates to applications or any primitive **UM** object and also to remove templates or change the order.





- To apply a template, click on the **Template Name** in the left pane and click the **Right Arrow**. The template appears in the right pane. You can also hold down the **Ctrl** key to select multiple templates.
- To remove a template already applied, click on the **Template Name** in the right pane and click the **Left Arrow**. The template disappears from the right pane.
- To change the order of the applied templates, click on the **Template Name** in the right pane and click the **Up Arrow** or **Down Arrow** to reposition the template.

You can apply multiple templates to applications or any primitive **UM** object. The order in which you attach templates determines the resolution of conflicting option values. The last instance of an option determines the value used. In the **UMM** GUI, this means the lowest option instance in the file.

3.7. Creating JMS Configurations

This section discusses the following topics.

- Configuring ConnectionFactories
- Configuring Factories
- Configuring Destinations

3.7.1. Configuring ConnectionFactories

- 1. Click on JMS in the object tree in the left pane and then right-click or click on the File Menu.
- 2. Select New and then select ConnectionFactory from the drop-down menu.
- 3. Enter the ConnectionFactory Name and click OK. The initial, blank ConnectionFactory window appears.

Figure 22. New ConnectionFactory Window

Ultra Messaging Manager		
File Edit Help		
🗂 root	newConnect	tionFactory UMQ UMP UMS
► Templates	Context	Name Value Ordering Parmid
Applications Applications Cillisers	Sources	
P I JMS	Receivers	
ConnectionFactory	WildCard	
newConnectionFactory TopicConnectionFactory	Receivers	
QueueConnectionFactory	ractory	
- C Destination		
- 🔝 Audit Log		
		Options Templates
		Permission
		Allow
		Ordering For All Contexts
		Comment
		Deny Allow 💌
	Graphical Vie	ew XML View

The listed ConnectionFactories, uJMSConnectionFactory, TopicConnectionFactory, and QueueConnectionFactory are pre-installed and facilitate running the compiled example JMS applications. See Ultra Messaging JMS Quick Start (.../QuickStart/jms-binary-quick-start.html).

4. If you wish to apply a template to the ConnectionFactory, click on the **Templates** button. The Templates window appears.



Figure 23. Templates List

See Applying Templates.

- 5. Configure the ConnectionFactory's Context.
 - If you wish to apply a template to the Context, click on the **Templates** button and apply the template(s).
 - Configure the options for the Context by clicking on the **Options** button and selecting the options. See *Creating Configuration Templates* for more on configuring options.
 - Configure the Sources. See Configuring Sources.
 - Configure the Receivers. See Configuring Receivers.
 - Configure any Wildcard Receivers. See Configuring Wildcard Receivers.
 - Configure the Factory options. See *Configuring Factories*.
 - Set the **Rule** attribute, if needed.
- 6. Configure options for the types of Destinations used with the ConnectionFactory. See Configuring Destinations.
 - For Destinations with the Type=UMS, click on the UMS tab and configure the appropriate options such as transport options.
 - For Destinations with the Type=UMP, click on the **UMP** tab and configure the appropriate options such as store configuration options.
 - For Destinations with the Type=UMQ, click on the UMQ tab and configure the appropriate options such as queue configuration options.

- 7. Right click on the ConnectionFactory name in the left pane or click on the File Menu. Then click Save All.
- 8. Click on the **XML View** tab to review the actual XML configuration for the ConnectionFactory. (Template options and values do not appear in this view.)

3.7.2. Configuring Factories

Follow the procedure below to configure Factory options for the ConnectionFactory.

1. Click on the Factory vertical tab and click on the Options button. The Factory Options list appears.

😡 Ultra Messaging Manager Co ectionFactory _ File Edit Help newCo default m ssage typ oroot Templates Application Users JMS use app heade Cont use index queu Sour default_topic_type WildCar 9 🗂 Co TopicCon Facto - 📑 Destir Select All Clear All OK Cancel

Figure 24. Factory Options Window

- 2. Select the Factory options by clicking in the checkbox for each desired option. Then click OK.
- 3. Enter the option values for the selected Factory options.

3.7.3. Configuring Destinations

Follow the procedure below to create Destinations and configure Destination options for the ConnectionFactory.

- 1. Click on JMS in the object tree in the left pane and then right-click or click on the File Menu.
- 2. Select New and then select Destination from the drop-down menu.
- 3. Enter the Destination Name and click OK. The initial, blank Destination window appears.

Figure 25. New Destination Window

The listed Destinations are pre-installed and facilitate running the compiled example JMS applications. See Ultra Messaging JMS Quick Start (../QuickStart/jms-binary-quick-start.html).

- 4. Configure the options for the Destination by clicking on the Options button and selecting the options.
- 5. Enter the option values for the selected Destination options.

3.8. Managing Users

You can authenticate the users of your applications by adding them to the User List and assigning applications to them. **UMM** does not serve configuration information to an application without the proper user authentication. See also *UM Manager Daemon*.

UMM comes with a Default User (username = default, password = default). The first time you start the **UMM** GUI, a prompt appears for the password of the Default User. When logged in as the Default User, you can change the Default User's password and other properties. You can also add other users. If you don't require extensive user authentication, you could assign all your applications to a single user.

Upon starting the UMM GUI for the first time, Informatica recommends that you either:

- · Change the password of the Default User, or
- Create a new administrative user, re-login as that new user and delete the Default User.

3.8.1. Changing the Default User Password

Use the following procedure to change the default user password.

- 1. Expand the Users object tree in the left pane.
- 2. Click on default. The Default User window appears.

Figure 26. Default User Window

Ultra Messaging Manager	
File Edit Help	
root root root root root root root voxop voxop voxop ustsConnection ustsConnection dots dots dots root source AuditLog	User Name default Password Change User Type Admin v Active v Applications • Applications • Add X Delete

- 3. Click on the Change button. The New Password dialog box appears.
- 4. Enter the new password and click on OK.

3.8.2. Managing Users

Follow the procedure below to add a new user.

- 1. Click on Users in the object tree in the left pane and then right-click or click on the File Menu.
- 2. Select New and then select User from the drop-down menu. The User Name dialog box appears.
- 3. Enter the User Name and click OK. The user profile dialog box appears.
- 4. Click in the **Password** entry box and enter the user's **Password**. (**UMM** hides the password after you save the user's record.)
- 5. Select the type of user, Normal or Admin.
- 6. Check the Active check box if the user is an active user.
- 7. Click on the Add button under the Applications entry box. A list of the configured applications appears.
- 8. Click on the applications this user has permission to run, holding down the **Ctrl** key to select multiple applications.
- 9. Select **Save** from the File Menu.

4. Using the UMM API

The **UMM** API 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. Programs that create application XML configurations store them in the **UMM** Configuration Database as shown in the diagram, *Architecture for Programming Application Configurations*. The UMM Daemon serves the XML configuration to **UM** applications.



Figure 27. Architecture for Programming Application Configurations

The following sample code provides a framework for creating an application configuration, test_application with two template_1 sets the file descriptor type option for a UM context.template_2 sets the resolver multicast address and port. Use of the application configuration is also authenticated with a username and password.

See <templates> (../Config/xmlconfigurationfiles.html#TEMPLATE-ELEMENTS) and <applications> (../Config/xmlconfigurationfiles.html#APPLICATIONS-ELEMENT) in the **UM** Configuration Guide for more information about XML configuration elements. See also *Creating Configuration Templates* and *Creating Application Configurations* for information on creating templates and application configurations using the **UMM** GUI.

```
package umm.api;
import java.io.CharArrayReader;
import java.util.ArrayList;
import java.util.logging.Level;
import java.util.logging.Logger;
import org.w3c.dom.Document;
import org.w3c.dom.Node;
import org.xml.sax.InputSource;
import umm.api.NameValue;
```

```
import umm.api.UMMAPI;
import umm.api.util.DOMUtil;
import umm.api.util.Util;
public class Example {
    String test_application = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n"
            + "<application name=\"test_application\" template=\"\">\n"
           + " <contexts order=\"deny,allow\">\n"
           + "<context rule=\"allow\" template=\"\">\n"
           + "<sources order=\"deny,allow\"/>\n"
           + " <receivers order=\"deny,allow\"/>\n"
           + "<wildcard-receivers order=\"deny,allow\"/>\n"
           + " </context>\n"
           + " </contexts>\n"
           + "<event-queues order=\"deny,allow\">\n"
           + " <event-queue rule=\"allow\"/>\n"
           + "</event-queues>\n"
           + "</application>";
    String template_1 = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n"
           + "<template name=\"template_1\">\n"
           + "<options type=\"context\">\n"
           + "<option default-value=\"epoll\" name=\"fd_management_type\"/>\n"
           + "</options>\n"
            + "</template>";
    String template_2 = "<?xml version=\"1.0\" encoding=\"UTF-8\"?>\n"
           + "<template name=\"template_2\">\n"
           + "<options type=\"context\">\n"
           + "<option default-value=\"224.9.10.11\" name=\"resolver_multicast_address\"/>\n"
           + "<option default-value=\"12965\" name=\"resolver_multicast_port\"/>\n"
            + "</options>\n"
            + "</template>";
    UMMAPI api;
   private void init (String machine, int port) throws Exception {
        //Create the UMMPI object. The constructor takes a machine name and port
        //which are used to connect to the UMM Daemon.
        api = new UMMAPI(machine, port);
        System.out.println("Init Success");
    }
   private void login(String userName, String password) throws Exception {
        //Logon to the UMM system. The login method must be called with a valid username
        //and password. The login user must be a UMM administrative user.
 if (api.login(userName, password)) {
            System.out.println("login Success");
            return;
        }
        System.out.println("login Failure");
        System.exit(-1);
    }
```

```
public Example(String args[]) {
   try {
       String machine = args[0];
       int port = Integer.parseInt(args[1]);
       String userName = args[2];
       String password = args[3];
       init(machine, port); //Connect to the UMM daemon
       login(userName, password); //Login to the daemon.
       ArrayList<NameValue> templateList = new ArrayList();
       // Need a list of templates to assign to the Application.
       NameValue nv = api.saveTemplate("template_1", template_1);
       if (nv == null) {
           System.out.println("saveTemplate Failure");
           System.exit(-1);
       }
       templateList.add(nv);
       nv = api.saveTemplate("template_2", template_2);
       if (nv == null) {
           System.out.println("saveTemplate Failure");
           System.exit(-1);
       }
       templateList.add(nv);
       String xml = setTemplateXML(templateList, test_application);
       if (xml == null) {
           System.out.println("setTemplateXML Failure");
           System.exit(-1);
       }
       //The template names must be set in the xml.
       if (api.saveApplication("test_application", xml) == null) {
           System.out.println("saveApplication Failure");
           System.exit(-1);
       }
       //The template names must be set in the Database.
       if (api.setTemplates("test_application", templateList) == false) {
           System.out.println("setTemplates Failure");
           System.exit(-1);
       }
       System.out.println(api.getConfig("test_application"));
   } catch (Exception ex) {
       Logger.getLogger(Example.class.getName()).log(Level.SEVERE, null, ex);
    }
```

```
api.logOff();
}
private String setTemplateXML(ArrayList<NameValue> list, String xml) {
    try {
        StringBuilder sb = new StringBuilder();
        //Build the template name string.
        for (NameValue nv : list) {
            sb.append(nv.getName() + ",");
        }
        // Remove the last ","
        if (sb.length() > 0) {
            sb.delete(sb.length() - 1, sb.length());
        }
        //Create a DOM document.
        CharArrayReader characterStream = new CharArrayReader(xml.toCharArray());
        InputSource is = new InputSource(characterStream);
        Document document = DOMUtil.createDocument(is);
        Node node = (Node) document.getDocumentElement();
        //Find the correct node to add the templates to.
        node = DOMUtil.findNode("contexts", node);
        if (node == null) {
            System.out.println("No contexts node");
            return null;
        }
        node = DOMUtil.findNode("context", node);
        if (node == null) {
            System.out.println("No context node");
            return null;
        }
        if (node.getAttributes().getNamedItem("template") != null) {
            node.getAttributes().getNamedItem("template").setNodeValue(sb.toString());
            return DOMUtil.getXML(document);
        }
    } catch (Exception ex) {
        ex.printStackTrace();
    }
    return null;
}
public static void main(String args[]) {
    new Example(args);
    System.exit(0);
}
```

}

5. UM Manager Daemon

The UMM Daemon serves license and configuration information to **UM** applications. A UMM Daemon requires either a umm.properties file or command line options. See ummd (../UMM/ummd.html) for information about the daemon's command line options. The UMM Daemon requires a license for either **UMP** or **UMQ**.

This section discusses the following topics.

- UMM Configuration
- · Connecting Your Applications to the UMM Daemon
- Configuring the UMM Database
- Securing UMM Daemon Communication with SSL

5.1. UMM Configuration

Follow the procedure below to configure UMM.

- 1. Edit the *umm.properties* file to specify your **UMM** database information, the daemon's port number and whether you will use SSL.
- 2. Edit the *ummd.bat or ummd.sh* file that starts UMM Daemon. The appropriate driver jar file name must be added to the classpath. Specify your certificate file or Java keystore information if using SSL certificates.
- 3. Edit the *umm.bat or umm.sh* file that starts the **UMM** GUI. The appropriate driver jar file name must be added to the classpath.
- 4. See *Connecting Your Applications to the UMM Daemon* to configure your applications with either environment variables or APIs so they can connect to UMM Daemon.
- 5. Optional. See Securing UMM Daemon Communication with SSL to configure SSL.

5.1.1. umm.properties

Contains database information such as the database username, password and driver. The simplest way to configure **UMM** is to edit this file to reflect your configuration database. The other two batch files or shell scripts automatically look to umm.properties for database information. This file, however, is optional. You could specify database information with options in the UMM Daemon and **UMM** GUI batch files or shell scripts. The umm.properties provided with installation appears below.

```
#database_username=system
#db_url=jdbc:oracle:thin:@//localhost:1521/xe
```

5.1.2. ummd.bat or ummd.sh

Specifies the needed information to run the UMM Daemon. By default, the ummd.bat/sh looks to umm.properties for database information. You can override the umm.properties file settings with ummd options. See *Manpage for ummd*. The ummd.bat or ummd.sh provided with installation appears below.

java -cp .;mysql-connector-java-5.0.8-bin.jar;UMMD_1.6.0_02.jar -Xms512m -Xmx1024m umm.ummd.Server

5.1.3. umm.bat or umm.sh

Specifies the needed information to run the UMM GUI. By default, the umm.bat/sh looks to umm.properties for database information. You can override the umm.properties file settings with ummd options. The umm.bat or umm.sh provided with installation appears below.

java -cp .;mysql-connector-java-5.0.8-bin.jar;UMM_1.jar; -Xms256m -Xmx1024m umm.gui.MainFrame

The same options available to the UMM Daemon (*Manpage for ummd*) are also available for the umm.bat batch file and the umm.sh shell script. The following two options are also available.

- -a UMM username
- -b UMM password

These options allow the **UMM** GUI user to skip the username and password dialog box when starting the GUI, but limits you to one user. See *Managing Users* for information about creating users and assigning passwords and applications to them.

5.2. Connecting Your Applications to the UMM Daemon

To use **UMM** with your installation of Ultra Messaging, you must use the following environment variable for every application/user combination.

```
export LBM_UMM_INFO=<application_name>:<user_name>:<password>@<ip>:<port>
```

For example:

```
export LBM_UMM_INFO=lbmrcv:default:default@10.29.3.95:21273
```

- 1. At start up, your UM application contacts the UMM Daemon at the given ip and port. You can specify multiple daemon locations in comma-separated format (@<ip>:<port>, <ip>:<port>, <ip>:<port>). Your application tries to connect in a round robin fashion if the daemon with index 0 fails.
- 2. If the connection succeeds, the UMM Daemon authenticates the connection with the user_name and password specified in the environment variable. (See *Managing Users* for information about creating users and assigning passwords and applications to them.)
- 3. Upon successful authentication, the UMM Daemon serves the license and configuration information associated with the user and application_name.

Note: Your UM application ignores any local licenses if it is configured for UMM. UMM can't be used to configure licenses for other UM daemons, such as umestored, tnwgd and lbmrd. These daemons must maintain their current local licensing.

4. The UMM Daemon closes the connection with your application.

You can specify the same information directly in your applications with the C API (../API/index.html), lbm_set_umm_info(), and the Java API (../JavaAPI/html/index.html) and .NET API
(../DotNetAPI/doc/Index.html), LBM.setUmmInfo(LBMUMMInfo). These must be the first calls made by your applications.

Note: The UMM Daemon currently only supports the configuration of sending and/or receiving applications. It does not support **UM** configuration options for other **UM** daemons, such as <code>umestored,tnwgd</code> and <code>lbmrd</code>. If running any of these daemons in the same environment, you should unset the LBM_UMM_INFO variable, start these other daemons and then reset LBM_UMM_INFO.

5.3. Configuring the UMM Database

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. This section discusses the following topics.

- Oracle Considerations
- MySQL Considerations

5.3.1. Oracle® Considerations

Informatica does not supply an Oracle database or any licensing to use Oracle. You can install an Oracle database either before or after installing UM. Drivers are not shipped with UM. The Oracle driver can be downloaded at http://www.oracle.com/technetwork/database/enterprise-edition/jdbc-10201-088211.html. The appropriate driver jar file name must be added to the classpath of the umm.bat/umm.sh and ummd.bat/ummd.sh scripts.

Configure Oracle with the following steps, which assume you have already installed UM.

- 1. Install Oracle and create a database. (You provide the UMM Daemon with the database name, username and password at start up. This user name and password should be kept secure.)
- 2. From the Oracle command line or database home page, log into your Oracle database.
- 3. From the Oracle command line or database home page, load /UMM/install_tables_oracle.sql. (This script installs the tables required by UMM.)
- 4. From the Oracle command line or database home page, load /UMM/oracle_templates.txt. (This script loads the example configuration templates for low latency (../Config/lowestlatency.html) and high throughput (../Config/examples.html#HIGHESTTHROUGHPUT). These templates appear in the UMM GUI under **Templates** in the object tree in the left pane.)
- 5. From the Oracle command line or database home page, load /UMM/oracle_application_jms.txt into the Application table. (This script loads the ConnectionFactories and Destinations required to run the example JMS applications. These templates appear in the UMM GUI under JMS/ConnectionFactory and JMS/Destination in the object tree in the left pane.)

You can now start the UMM Daemon on the command line or with the ummd.bat batch file or the ummd.sh shell script.

5.3.2. MySQL[™] Considerations

Informatica does not supply MySQL or any licensing to use MySQL. You can install MySQL either before or after installing UM. Drivers are not shipped with UM. The MySQL driver can be downloaded at http://dev.mysql.com/downloads/connector/j/5.0.html. The appropriate driver jar file name must be added to the classpath of the umm.bat/umm.sh and ummd.bat/ummd.sh scripts.

Configure MySQL with the following steps, which assume you have already installed UM.

- 1. Install MySQL.
- 2. Create a database. (You provide the UMM Daemon with the database name, username and password at start up. This user name and password should be kept secure.)
- 3. Log into the MySQL with connect db_name.
- 4. From the MySQL command prompt, run source /UMM/install_tables_mysql.sql. (This script installs the tables required by UMM.)
- 5. From the MySQL command prompt, run LOAD DATA INFILE 'mysql_templates.txt' INTO TABLE TEMPLATE; (This script loads the example configuration templates for low latency (../Config/lowestlatency.html) and high throughput (../Config/examples.html#HIGHESTTHROUGHPUT). These templates appear in the UMM GUI under Templates in the object tree in the left pane.)
- 6. From the MySQL command prompt, run LOAD DATA INFILE 'mysql_application_jms.txt' INTO TABLE APPLICATION;. (This script loads the ConnectionFactories and Destinations required to run the example JMS applications. These templates appear in the UMM GUI under JMS/ConnectionFactory and JMS/Destination in the object tree in the left pane.)

You can now start the UMM Daemon on the command line or with the ummd.bat batch file or the ummd.sh shell script.

5.4. Securing UMM Daemon Communication with SSL

UMM allows you to secure the TCP transmission of **UMM** credentials and **UM** license keys between the UMM Daemon and **UM** applications using Secure Sockets Layer (SSL). You can implement a certificate-less, Transport Layer Security (TLS) option or provide your own certificate to authenticate the UMM Daemon. The following procedure explains how to secure UMM Daemon communications.

1. In the *umm.properties*, set the secure option to true.

```
secure=true
```

If you run the UMM Daemon from the command line, use -s true.

2. In the Connecting Your Applications to the UMM Daemon, pre-pend the application information with ssl:.

```
export LBM_UMM_INFO=ssl:lbmrcv:default:default@10.29.3.95:21273
```

3. To specify a certificate, insert the certificate path and filename as shown below. Be sure to also configure *ummd.bat or ummd.sh* with your keystore.

export LBM_UMM_INFO=ssl_cert_file=path/filename:lbmrcv:default:default@10.29.3.95:21273

A password for the certificate file is optional and would be specified as follows.

```
export LBM_UMM_INFO=ssl_cert_file=path/filename:ssl_cert_file_password=psswrd:
lbmrcv:default:default@10.29.3.95:21273
```

You can specify the same information directly in your applications with the C API (../API/index.html), lbm_set_umm_info(), and the Java API (../JavaAPI/html/index.html) and .NET API
(../DotNetAPI/doc/Index.html), LBM.setUmmInfo(LBMUMMInfo). These must be the first calls made by your applications.

6. Manpage for ummd

ummd

Name

 $\mathsf{ummd} - \mathsf{UM} \mathsf{Manager} \mathsf{Daemon}$

Synopsis

ummd [-d JDBC Driver] [-h] [-n db username] [-p port] [-P properties filename] [-s true/false] [-u db url] [-x db password]

Description

ummd provides UM Manager services.

You can identify the database used to store configuration information with the -u *db url* option. This option accepts JDBC format for a MySQL database (i.e. jdbc:mysql://10.29.3.112:3306/name) or an Oracle database (i.e. jdbc:oracle:thin:@//localhost:1521/xe). You can also set the database's username with the -n option, the database password with the -x, and the database driver with the -d. All options set on the ummd command line override any settings in the umm.properties file.

You can set an alternate properties file with the -P where you can also specify database information. The default properties file is umm.properties.

Command line help is available with -h.

You can set the port for ummd with the -p option. The default is 15701.

The -s option let you specify the use of SSL encryption for communication between ummd and your UM applications.

Exit Status

The exit status from ummd is 0 for success and some non-zero value for failure.