



Informatica® RulePoint
6.1

Administrator Guide

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(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 © Sun Microsystems. All rights reserved. Copyright © RSA Security Inc. All Rights Reserved. Copyright © Ordinal Technology Corp. All rights reserved. Copyright © Aandacht c.v. All rights reserved. Copyright Genivia, Inc. All rights reserved. Copyright Isomorphic Software. All rights reserved. Copyright © Meta Integration Technology, Inc. All rights reserved. Copyright © Intalio. All rights reserved. Copyright © Oracle. All rights reserved. Copyright © Adobe Systems Incorporated. All rights reserved. Copyright © DataArt, Inc. All rights reserved. Copyright © ComponentSource. All rights reserved. Copyright © Microsoft Corporation. All rights reserved. Copyright © Rogue Wave Software, Inc. All rights reserved. Copyright © Teradata Corporation. All rights reserved. Copyright © Yahoo! Inc. All rights reserved. Copyright © Glyph & Cog, LLC. All rights reserved. Copyright © Thinkmap, Inc. All rights reserved. Copyright © Clearpace Software Limited. All rights reserved. Copyright © Information Builders, Inc. All rights reserved. Copyright © OSS Nokalva, Inc. All rights reserved. Copyright Edifecs, Inc. All rights reserved. Copyright Cleo Communications, Inc. All rights reserved. Copyright © International Organization for Standardization 1986. All rights reserved. Copyright © ej-technologies GmbH. All rights reserved. Copyright © Jaspersoft Corporation. All rights reserved. Copyright © is International Business Machines Corporation. All rights reserved. Copyright © yWorks GmbH. All rights reserved. Copyright © Lucent Technologies. All rights reserved. Copyright (c) University of Toronto. All rights reserved. Copyright © Daniel Veillard. All rights reserved. Copyright © Unicode, Inc. Copyright IBM Corp. All rights reserved. Copyright © MicroQuill Software Publishing, Inc. All rights reserved. Copyright © PassMark Software Pty Ltd. All rights reserved. Copyright © LogiXML, Inc. All rights reserved. Copyright © 2003-2010 Lorenzi Davide, All rights reserved. Copyright © Red Hat, Inc. All rights reserved. Copyright © The Board of Trustees of the Leland Stanford Junior University. All rights reserved. Copyright © EMC Corporation. All rights reserved. Copyright © Flexera Software. All rights reserved. Copyright © Jinfonet Software. All rights reserved. Copyright © Apple Inc. All rights reserved. Copyright © Telerik Inc. All rights reserved. Copyright © BEA Systems. All rights reserved. Copyright © PDFlib GmbH. All rights reserved. Copyright © Orientation in Objects GmbH. All rights reserved. Copyright © Tanuki Software, Ltd. All rights reserved. Copyright © Ricebridge. All rights reserved. Copyright © Sencha, 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 © 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 (©) 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 1996-2013, 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 (©) 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 © 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 © 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 © 2002 Ralf S. Engelschall, Copyright © 2002 The OSSP Project Copyright © 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 © 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 © 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> and at <http://www.eclipse.org/org/documents/edl-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/hsqldbLicense.html>, <http://httpunit.sourceforge.net/doc/license.html>, <http://jung.sourceforge.net/license.txt>, 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://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/ODBC/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>, <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>, <http://jsoncpp.sourceforge.net/LICENSE>, <http://jdbc.postgresql.org/license.html>, <http://protobuf.googlecode.com/svn/trunk/src/google/protobuf/descriptor.proto>, <https://github.com/rantav/hector/blob/master/LICENSE>, <http://web.mit.edu/Kerberos/krb5-current/doc/mitK5license.html>, <http://jibx.sourceforge.net/jibx-license.html>, and <https://github.com/lyokato/libgeohash/blob/master/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 new BSD License (<http://opensource.org/licenses/BSD-3-Clause>), the MIT License (<http://www.opensource.org/licenses/mit-license.php>), the Artistic License (<http://www.opensource.org/licenses/artistic-license-1.0>) and the Initial Developer's Public License Version 1.0 (<http://www.firebirdsql.org/en/initial-developer-s-public-license-version-1-0/>).

This product includes software copyright © 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 product includes software Copyright (c) 2013 Frank Balluffi and Markus Moeller. All rights reserved. Permissions and limitations regarding this software are subject to terms of the MIT license.

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; 6,823,373; 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; 7,774,791; 8,065,266; 8,150,803; 8,166,048; 8,166,071; 8,200,622; 8,224,873; 8,271,477; 8,327,419; 8,386,435; 8,392,460; 8,453,159; 8,458,230; and RE44,478, 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.

Publication Date: 2018-07-27

Table of Contents

Preface	11
Informatica Resources.	11
Informatica My Support Portal.	11
Informatica Documentation.	11
Informatica Web Site.	11
Informatica How-To Library.	11
Informatica Knowledge Base.	12
Informatica Support YouTube Channel.	12
Informatica Marketplace.	12
Informatica Velocity.	12
Informatica Global Customer Support.	12
 Chapter 1: Introduction to RulePoint.....	13
RulePoint Overview.	13
RulePoint Example.	14
RulePoint User Interface.	14
RulePoint Design-Time Environment.	14
RulePoint Run-Time Environment.	14
Understanding a Topology.	15
Host.	15
Node.	15
Host Agent.	15
Application Services.	15
System Services.	16
Default Topology.	16
Objects Deployment.	17
Process Flow in a Default Topology.	18
Scalability and High Availability in RulePoint.	18
Vertical and Horizontal Scalability	18
Partitioning.	19
High Availability	19
User Authentication.	19
Dashboard.	19
 Chapter 2: Managing Your Account.....	20
Managing Your Account Overview.	20
Starting RulePoint.	20
Setting Up RULEPOINT_HOME Environment Variable.	21
Starting and Stopping the H2 Database.	21
Starting and Stopping the RulePoint Host Agent Instance.	22

Starting and Stopping RulePoint Design-Time Instance.	23
Starting and Stopping RulePoint Topology Instance.	24
Logging In to RulePoint.	25
Configuring User Preferences.	25
Changing Your Password.	25
Changing the Splash Screen Configuration	25
Changing the Date and Time Format.	26
Chapter 3: Using RulePoint User Interface.....	27
Using RulePoint User Interface Overview.	27
Administration Tab.	28
User Management View.	28
User View.	28
Role View.	29
Import View.	29
Export View.	30
Topology View.	30
Host.	30
Node.	31
Application Services.	31
System Services.	32
Chapter 4: User Management.....	34
User Management Overview.	34
Default Users, Roles, and Privileges.	34
Privilege Types for RulePoint Users.	35
Access Control Lists in RulePoint.	35
Permission Types.	36
ACL Permission Rules	36
Evaluating ACL Rule Permission Requirements.	37
Authentication.	37
LDAP Authentication	37
Configuring LDAP Authentication for RTAM.	40
Managing Users and Roles.	42
Best Practices for User Management.	45
Chapter 5: Topology Management.....	46
Topology Management Overview.	46
Topology Planning.	46
Topology Design.	47
Topology Design Considerations.	48
Configuring a Topology.	49
Topology Properties.	50

Editing a Topology.	51
Manually Setting Properties in Configuration Files.	51
Chapter 6: Managing Hosts and Nodes.	53
Managing Hosts and Nodes Overview.	53
Host.	53
Configuring a Host.	54
Adding a Host.	54
Editing a Host.	54
Viewing Hosts.	55
Deleting a Host.	56
Node.	56
Configuring a Node.	56
Adding a Node.	57
Editing a Node.	57
Viewing Nodes.	58
Deleting a Node.	58
Chapter 7: Managing Application Services.	60
Application Services Overview.	60
Service Controller.	60
Configuring a Source Controller.	61
Source Controller Properties.	61
Creating a Source Controller in Standalone Mode.	62
Creating a Source Controller in High-Availability Mode.	62
Editing a Source Controller.	63
Viewing a Specific Source Controller.	63
Viewing All Source Controllers.	64
Deleting a Source Controller.	64
Configuring a Responder Controller.	65
Responder Controller Properties.	65
Creating a Responder Controller in Standalone Mode.	66
Creating a Responder Controller in High-Availability Mode.	67
Editing a Responder Controller.	67
Viewing a Specific Responder Controller.	68
Viewing All Responder Controllers.	68
Deleting a Responder Controller.	69
Event Processor.	69
Configuring an Event Processor.	69
Event Processor Properties.	69
Creating an Event Processor in Standalone Mode.	70
Creating an Event Processor in High-Availability Mode.	71
Editing an Event Processor.	71

Viewing a Specific Event Processor.	72
Viewing All Event Processors.	72
Deleting an Event Processor.	72
Activity Manager.	73
Configuring the Activity Manager.	73
Activity Manager Properties.	73
Adding an Activity Manager.	75
Editing an Activity Manager.	75
Deleting an Activity Manager.	76
Creating Multiple Instances of Activity Manager.	76
Chapter 8: Managing System Services.....	78
System Services Overview.	78
Grid Manager.	78
Configuring a Grid Manager.	79
Grid Manager Configuration Properties.	79
Adding a Grid Manager to a Default Topology	81
Editing a Grid Manager.	81
Viewing a Specific Grid Manager.	82
Viewing All Grid Managers.	82
Deleting a Grid Manager.	82
UM lbmrd.	83
Configuring the UM lbmrd.	83
Adding a UM lbmrd to a Default Topology.	83
Editing a UM lbmrd.	84
Viewing a Specific UM lbmrd.	84
Viewing All UM lbmrd.	84
Deleting a UM lbmrd.	85
UM Store.	85
Configuring a UM Store.	85
Adding a UM Store to a Default Topology.	85
Editing a UM Store.	86
Viewing a Specific UM Store.	86
Viewing All UM Stores.	87
Deleting a UM Store.	87
Chapter 9: High Availability.....	88
High-Availability Overview in RulePoint	88
Achieving High Availability.	89
Workflow for Configuring High Availability.	89
Introduction to Resiliency in RulePoint.	90
Failure Scenarios and Failover Actions for the Run-Time Components.	90
Failover in the Grid Manager.	91

Availability in the Grid Manager.	92
Failover of the Source Controller and the Responder Controller.	92
Availability of the Source Controller.	93
Availability of the Responder Controller.	93
Failover of the Event Processor.	94
Availability of the Event Processor.	95
Best Practices for Achieving High Availability.	96
Chapter 10: Managing Deployment	98
Deployment Overview.	98
State of RulePoint Objects.	100
Deployment Tasks.	100
Deployment Options.	102
Deployment Workflow.	102
Use Case: Deployment Types and State Transitions.	103
Undeploying Running Objects.	104
Error Management During Deployment.	105
Deployment Considerations and Best Practices.	105
Deploying RulePoint Objects.	106
Prerequisites for Deploying Objects.	106
Deploying a Single Source.	107
Deploying Multiple Sources.	107
Deploying a Rule.	107
Deploying Multiple Rules.	108
Deploying a Single Responder.	108
Deploying Multiple Responders.	108
Deploying All Objects Simultaneously.	109
Deployment Policy for Templates.	109
Create a Deployment Policy for a Template	109
Creating a Deployment Policy for Multiple Templates.	110
Update a Deployment Policy	110
Delete a Deployment Policy.	110
Redeploying the RulePoint Objects	110
Redeploying a Source.	111
Redeploying Multiple Sources.	111
Redeploying a Rule.	111
Redeploying Multiple Rules.	112
Redeploying a Responder.	112
Redeploying Multiple Responders.	112
Redeploying All Objects Simultaneously.	113
Undeploying the RulePoint Objects.	113
Undeploying a Source.	113
Undeploying Multiple Sources.	113

Undeploying a Rule.	113
Undeploying Multiple Rules.	114
Undeploying a Responder.	114
Undeploying Multiple Responders.	114
Undeploying All Objects Simultaneously.	114
Reassigning the RulePoint Objects.	115
Reassigning a Source.	115
Reassigning Multiple Sources.	115
Reassigning a Responder.	115
Reassigning Multiple Responders.	116
Reassigning All Objects Simultaneously.	116
Chapter 11: Dashboard.....	117
Overview of Dashboard.	117
Dashboard Interface.	117
Using the Dashboard.	118
Metrics View.	118
Events View.	119
Logs View.	121
Objects View for an Application Service.	122
Source Controller.	122
Event Processor.	124
Responder Controller.	126
Task Types for Objects on the Dashboard.	126
Task Types for Objects in Source Controller.	127
Task Types for Objects in Event Processor.	127
Task Types for Objects in Responder Controller.	128
Task Types for Objects at Topology and Component Level.	129
Use Case: Understanding Dashboard Functions and Troubleshooting.	129
Scenario 1: Viewing Topics and Events for a Source.	129
Scenario 2: Viewing Responses.	130
Scenario 3: Checking Rule Activation.	130
Scenario 4: Purging Events and Alert Details of Objects	130
Scenario 5: Using Tracing for Troubleshooting.	131
Scenario 6: Creating Events for Troubleshooting.	131
Scenario 7: Copying Events for Troubleshooting.	132
Chapter 12: Object Import and Export.....	133
Object Import and Export Overview.	133
Import.	133
Uploading a File.	134
Importing a File.	134
Deleting an Imported File and History.	134

Export.	135
Exporting Selected Objects.	135
Exporting All Objects.	136
Downloading a File.	136
Deleting an Exported File and History.	137
Chapter 13: Markers.	138
Markers Overview.	138
Source Markers.	138
Marker Rules.	139
System Marker Properties.	139
Configuring a Marker.	140
Best Practices for Using Markers.	140
Chapter 14: Log Management.	141
Log Management Overview.	141
RulePoint Logs.	141
Log File Configurations.	142
Viewing the Log Files.	142
Log File Location.	142
Log Format.	143
Chapter 15: Licenses.	145
Licenses Overview.	145
License Validation.	145
Appendix A: Error Codes.	146
Design-Time Error Codes.	146
Interaction Error Codes.	149
Import Error Codes.	150
Export Error Codes.	151
Security Error Codes.	151
ACL Error Codes	152
Run-Time Error Codes.	153
Appendix B: Glossary.	160
Index.	163

Preface

The *RulePoint Administrator Guide* is written for administrators. This guide contains information you need to configure the run-time components and administer RulePoint. It also provides deployment information for RulePoint objects.

This guide assumes that you have a basic working knowledge of RulePoint and are familiar with the RulePoint concepts.

Informatica Resources

Informatica My Support Portal

As an Informatica customer, you can access the Informatica My Support Portal at <http://mysupport.informatica.com>.

The site contains product information, user group information, newsletters, access to the Informatica customer support case management system (ATLAS), the Informatica How-To Library, the Informatica Knowledge Base, Informatica Product Documentation, and access to the Informatica user community.

Informatica Documentation

The Informatica Documentation team takes every effort to create accurate, usable documentation. If you have questions, comments, or ideas about this documentation, contact the Informatica Documentation team through email at infa_documentation@informatica.com. We will use your feedback to improve our documentation. Let us know if we can contact you regarding your comments.

The Documentation team updates documentation as needed. To get the latest documentation for your product, navigate to Product Documentation from <http://mysupport.informatica.com>.

Informatica Web Site

You can access the Informatica corporate web site at <http://www.informatica.com>. The site contains information about Informatica, its background, upcoming events, and sales offices. You will also find product and partner information. The services area of the site includes important information about technical support, training and education, and implementation services.

Informatica How-To Library

As an Informatica customer, you can access the Informatica How-To Library at <http://mysupport.informatica.com>. The How-To Library is a collection of resources to help you learn more

about Informatica products and features. It includes articles and interactive demonstrations that provide solutions to common problems, compare features and behaviors, and guide you through performing specific real-world tasks.

Informatica Knowledge Base

As an Informatica customer, you can access the Informatica Knowledge Base at <http://mysupport.informatica.com>. Use the Knowledge Base to search for documented solutions to known technical issues about Informatica products. You can also find answers to frequently asked questions, technical white papers, and technical tips. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team through email at KB_Feedback@informatica.com.

Informatica Support YouTube Channel

You can access the Informatica Support YouTube channel at <http://www.youtube.com/user/INFASupport>. The Informatica Support YouTube channel includes videos about solutions that guide you through performing specific tasks. If you have questions, comments, or ideas about the Informatica Support YouTube channel, contact the Support YouTube team through email at supportvideos@informatica.com or send a tweet to @INFASupport.

Informatica Marketplace

The Informatica Marketplace is a forum where developers and partners can share solutions that augment, extend, or enhance data integration implementations. By leveraging any of the hundreds of solutions available on the Marketplace, you can improve your productivity and speed up time to implementation on your projects. You can access Informatica Marketplace at <http://www.informaticamarketplace.com>.

Informatica Velocity

You can access Informatica Velocity at <http://mysupport.informatica.com>. Developed from the real-world experience of hundreds of data management projects, Informatica Velocity represents the collective knowledge of our consultants who have worked with organizations from around the world to plan, develop, deploy, and maintain successful data management solutions. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

Informatica Global Customer Support

You can contact a Customer Support Center by telephone or through the Online Support.

Online Support requires a user name and password. You can request a user name and password at <http://mysupport.informatica.com>.

The telephone numbers for Informatica Global Customer Support are available from the Informatica web site at <http://www.informatica.com/us/services-and-training/support-services/global-support-centers/>.

CHAPTER 1

Introduction to RulePoint

This chapter includes the following topics:

- [RulePoint Overview, 13](#)
- [RulePoint User Interface, 14](#)
- [Understanding a Topology, 15](#)
- [Default Topology, 16](#)
- [Objects Deployment, 17](#)
- [Process Flow in a Default Topology, 18](#)
- [Scalability and High Availability in RulePoint, 18](#)
- [User Authentication, 19](#)
- [Dashboard, 19](#)

RulePoint Overview

RulePoint is a complex event processing solution that delivers real-time operational business intelligence to users within an enterprise.

You can create business rules in RulePoint to identify hidden patterns in data and correlate real-time events that negatively impact your business. In RulePoint, you can monitor diverse and disparate data or event sources, including event streams, sensors, communications systems, message queues, web services, RSS feeds, databases, and flat files.

You use the data sources to connect to external systems to extract data. The sources convert the extracted data into meaningful events. Sources then logically group events and publish them to topics. You need to configure rules to analyze and detect events associated with the topic. When events match the conditions specified in the rule, RulePoint notifies a response to the appropriate user through responders.

A key RulePoint advantage is that it helps you discover operational changes as they occur and the opportunity to respond to important events immediately.

RulePoint can process large volumes of data in a highly available and fault-tolerant environment. You can scale services and share objects across multiple machines. The high-availability functionality helps minimize service downtime due to unexpected failures.

RulePoint Example

You can use RulePoint to monitor banking account transactions to identify fraud in real time and respond to important events as they occur.

Use RulePoint to collect related information from multiple data sources and apply rules to identify different patterns of misuse in real time. RulePoint delivers targeted alerts and notifications across multiple channels, including Real-Time Alert Manager, instant messages, and email. Notifications across these channels enable the responsible individuals and groups to immediately begin an investigation.

RulePoint User Interface

The user interface in RulePoint is browser based, and it has a distinct design-time and run-time environment to manage tasks.

You need to create the primary and secondary RulePoint objects in the design-time environment. The application services run the deployed objects in the run-time environment.

You can also manage users, configure and scale the topology components, administer the run-time environment, import and export RulePoint objects, and perform other administration-related tasks.

Use the dashboard to view the run-time activity and trends of the RulePoint objects. You can monitor events continuously and gain visibility into the system health. You can also view the configuration, availability, and performance of application services on the dashboard.

RulePoint Design-Time Environment

RulePoint requires you to initially create the RulePoint design objects.

The primary objects are the source, rule, and responder. Supporting objects include analytics, topics, watchlists, connections, templates, and response. For information about creating objects, see the *RulePoint User Guide*. After you create objects in the design time, you deploy the objects into the corresponding application services to process the rules.

RulePoint Run-Time Environment

Application services run the deployed rule-processing objects in the run-time environment to generate alerts. Application services include the source controller, event processor, and responder controller.

A source controller pulls data from configured external event sources, pre-processes an inbound event, and forwards the event to an event processor. An event processor processes events based on user-defined rules. When an event matches the rule conditions, the event processor produces responses into a responder controller. A responder controller delivers responses to a configured external response target.

The application services run on nodes, which are Java virtual machine (JVM) processes. System services manage all the RulePoint run-time components that run on the host machine.

You can configure the application and system services for scalability and high availability. You can see the activity of each of the components on the dashboard. The run-time components distributed across nodes use Informatica Ultra Messaging (UM) to communicate with each other.

Understanding a Topology

The topology is the fundamental administrative unit in RulePoint. The topology supports the administration of the distributed services.

A topology is a group of application services, system services, nodes, and hosts that you can group into folders based on administrative ownership. A topology constitutes the run-time environment in RulePoint.

Host

A host is a physical machine where you install RulePoint. The host consists of nodes, application services, and system services.

On each host, you can configure one or more nodes, grid manager, UM lbmrd, and UM stores. A grid manager manages all nodes, application services, and system services configured in the host.

Node

A node is a JVM process that manages the application services. Application services and system services communicate with the grid manager through the node.

Host Agent

A host agent runs on each host in a topology and manages communication between the grid manager and the nodes configured on a host.

Application Services

The application services run on nodes, and include the source controller, event processor, responder controller, and activity manager.

Source Controller

The source controller manages the lifecycle of the source and its supporting objects, such as connections and topics. A source controller transforms messages between an inbound data source to the event processor.

The source controller pulls data from configured data sources and transforms data into events based on a set of criteria that you specify. It then publishes the events and groups them into categories as topics.

Event Processor

The event processor manages the lifecycle of rules and dependent objects, such as topics, analytics, watchlists, connections, and responses deployed in it.

The event processor correlates an incoming event from a source controller with past events based on rules that you defined for that event topic. Event processor processes events based on the rules, and when events match rule conditions, it executes the response specified in the rule.

Responder Controller

The responder controller manages the lifecycle of responders and their related objects, such as responses and connections. The responder controllers consume activations from the event processor to the outbound data sources.

An activation is based on a configured set of criteria that you specify. The responder controller pushes data to a responder and transforms it into a response. The responder controller manages the lifecycle of the responder.

Activity Manager

The activity manager records and stores the activity of the application services, such as the object execution information and service level aggregate statistics. It displays the data recorded on the dashboard.

System Services

System services are physical processes that run on the host machine. System services include the grid manager, UM store, and UM lbmrd.

Grid Manager

The grid manager controls the functioning of the application and system services.

The grid manager manages deployment of objects into the application services and handles all interactions between the design-time and run-time environment.

UM Store

The UM store is based on an asynchronous messaging model called Ultra Messaging Persistence. The UM store enables guaranteed delivery for data exchange across all services within RulePoint. The application services use the UM store to persist events.

UM lbmrd

Ultra Messaging Latency Busters Messaging Resolver Daemon (UM lbmrd) provide the address resolver capability for data exchange across all services in RulePoint.

The application services use unicast topic resolution to create subscribers and publishers to the topic. The store configuration file consists of the unicast topic resolution daemon (lbmrd) properties.

Default Topology

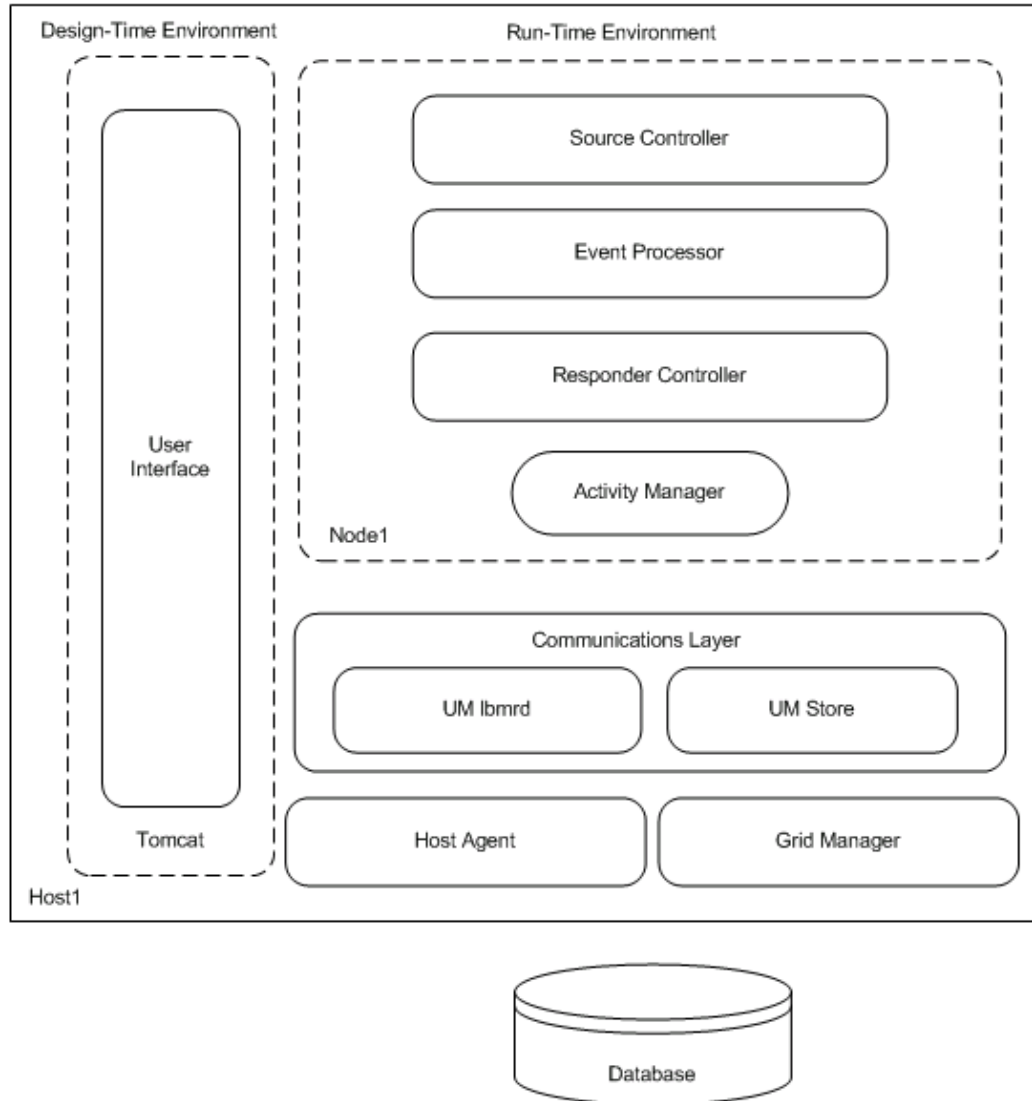
A default topology consists of a single node that runs on a host, along with the required system and application services. By default, when you install RulePoint, it creates the default topology. In a default topology, the Tomcat application service that runs the design time, is in the same machine as the run-time machine.

All the run-time components, such as the application services and the system services, run on a single host.

Application services include an instance each of the source controller, responder controller, event processor, and activity manager that run on a node. System services include an instance each of the grid manager, UM

store, and UM lbmrd service that run on the host. All the services operate in standalone mode. The run-time database stores all the topology configurations. You can, however, choose separate databases during installation for the run-time and design-time configurations.

The following figure shows the default topology, which consists of a single-host, single-node setup:



Objects Deployment

You create objects and deploy them to application services to process the events.

When you deploy objects in a default topology, sources and supporting objects are deployed to the source controller, responders and supporting objects are deployed into the responder controller, while rules and supporting objects are deployed in the event processor. After you deploy objects, the source controller begins to fetch events, rule processor processes events, and responder controller dispatches alerts.

Process Flow in a Default Topology

After you create and deploy the objects, the application services process the objects against the configured rules and generates alerts.

The following process describes the deployment flow in RulePoint:

1. In the design time, create a project with a set of primary and supporting RulePoint objects.
2. Deploy the sources, rules, and responders.
3. The grid manager initiates the deployment of objects to the corresponding application services. Sources are deployed to the source controller, rules to the event processor, and responders to the responder controller.
4. The grid manager sends a success status when it deploys an object. The state of objects changes from draft to deployed. If the deployment fails, the grid manager sends an error message.
5. The source controller connects to the external system by using the corresponding protocol handler, and publishes the events on topics subscribed to it. Subscriptions are driven by the DRQL rule.
6. The source controller routes the events to the event processor. The event processor applies processing logic based on the rules to incoming events as they are associated with the topic.
7. When the event matches the rule conditions, the rule sends a response into the responder controller. The responses contain details of the error or the event.
8. The activity manager records the management information of all the services.
9. The administration screen dashboard on the user interface displays the run-time activity of all the deployed objects.

For a high-level workflow of all configurations in the run-time environment, see the *RulePoint Getting Started Guide*.

Scalability and High Availability in RulePoint

You can configure the RulePoint environment for vertical and horizontal scalability, partitioning, high availability, and failover to effectively manage your business operations. You can configure the components to work in standalone mode or high-availability mode.

You can scale the application and system services in a default topology. You can configure multiple instances of source controllers, event processors, and responder controllers to run on a single node for small-scale processing purposes. If you want to perform large-scale processing, you can configure different combinations of source controllers, event processors, and responder controllers to run on multiple nodes, and you can distribute nodes across multiple hosts. You can have n application services on one node, n nodes on one host, and n hosts in one topology. You can achieve load balancing by using the appropriate partitioning strategy. When you deploy objects, you need to partition the events and rules across corresponding application services.

Vertical and Horizontal Scalability

When the overall volume of data increases, you can scale the topology vertically or horizontally based on your requirements.

If you want to increase the working memory of a specific application service, you can vertically scale the host to handle the increased load. You can add more CPUs to an existing host, and configure the application

service to take advantage of the additional memory. When you extend the memory and increase the CPU, you increase the performance and throughput of the application service.

You can also horizontally scale the application services to accommodate increased event or rule volume, or for overall availability. You can configure multiple instances of source controllers, event processors, and responder controllers. Each node might contain one or more of the application services. The application services on a node can be of similar type or mixed, as required. You can partition sources, rules, and responders across corresponding application services across multiple nodes. You can also scale the system services to include multiple instances.

Partitioning

You can use partitioning to make RulePoint a highly scalable environment for high-volume transactional processing.

During deployment, you can partition the service workload of the sources, rules, and responders across specific application services. You can dedicate nodes for specific services to manage RulePoint more effectively. You can configure specific nodes for high availability and failure isolation.

High Availability

You can configure the topology to be highly availability to eliminate single points of failure.

If your business requires high availability, you can replicate the application services and the system services as primary and backup instances. You can distribute these services across nodes to ensure that even if one node goes down, the backup instances are still available and continue to generate events, process rules, and dispatch alerts. Failover of each of the run-time components switches over from the primary to the backup instance seamlessly.

User Authentication

When you log in, RulePoint authenticates your credentials by using the local or remote authentication provider. The configured access control list for each object and the groups to which you have assigned the user in the authentication provider control user access.

When you log in as a local user, RulePoint authenticates your user name and password against the user accounts in the topology configuration database. When you log in to RulePoint as an LDAP user, RulePoint passes your user name and password to the external directory service for authentication.

Dashboard

The dashboard in RulePoint is a real-time user interface, showing a graphical presentation of the trends of objects and RulePoint performance.

The dashboard provides real-time visibility to the activity and state of the configured application services and the objects deployed on the application services. Use the dashboard to view the summary information about RulePoint event processing for a set time line. The dashboard also displays functions to stop or start an object, and purge object metrics. You can also enable rule tracing to get a visual event correlation for a particular rule.

CHAPTER 2

Managing Your Account

This chapter includes the following topics:

- [Managing Your Account Overview, 20](#)
- [Starting RulePoint, 20](#)
- [Logging In to RulePoint, 25](#)
- [Configuring User Preferences, 25](#)

Managing Your Account Overview

Manage your account to change your password or edit the user name.

If you have a local user account, you can change your password and name with the User Preferences application. A RulePoint user must have super user privileges to perform the administration tasks.

Starting RulePoint

Before you log in to RulePoint, you must start each RulePoint instance.

1. Host agent
2. Topology
3. Design time

If you register the services during installation, you can also use the following Windows Services to start or stop the following RulePoint instances:

- RulePoint HostAgent
- RulePoint Topology
- RulePoint Design Time

If you selected the database as H2 during installation, you need to start the H2 database. You can use the command prompt to start the database. If you register for the Windows services during RulePoint installation on Windows, you can start the RulePoint H2 DB service from the Windows services.

Setting Up RULEPOINT_HOME Environment Variable

Before you start RulePoint instances, create the RULEPOINT_HOME environment variable. Set RULEPOINT_HOME to the location of the RulePoint installation directory.

The following file path is the default location of the RulePoint installation directory on the supported platforms:

- **Windows.** C:\RulePoint_6.1.
- **Linux.** <RulePointInstallationDir>/RulePoint_6.1.
- **Solaris.** <RulePointInstallationDir>/RulePoint_6.1.
- **AIX.** <RulePointInstallationDir>/RulePoint_6.1.

Starting and Stopping the H2 Database

During installation, if you select the database type as H2 for the design-time and run-time repository, you need to start the H2 database before you start the RulePoint instances.

Starting and Stopping the H2 Database on Windows

Use the `startDB.bat` command to start the H2 database on Windows.

Note: If you have registered services for H2, you can use the RulePoint H2 DB service from the Windows services to start or stop the database.

RulePoint installs `startDB.bat` in the following directory by default:

<RULEPOINT_HOME>/bin

1. Go to the directory where `startDB.bat` is located.
2. Open the command prompt as an administrator or user with all the required permissions. Run the following command to start the H2 database:

```
startDB.bat
```

Enter the following command to stop the H2 database:

```
stopDB.bat
```

Starting and Stopping the H2 Database on Unix

You can use the `startDB.sh` command to start the H2 database on Unix.

Note: If you have registered services for H2, you can use the RulePoint H2 DB service from the Windows services to start or stop the database.

RulePoint installs `startDB.sh` in the following directory by default:

<RULEPOINT_HOME>/bin

1. Go to the directory where `startDB.sh` is located.
2. Open the command prompt as an administrator or user with all the required permissions. Run the following command to start the H2 database:

```
startDB.sh
```

Enter the following command to stop the H2 database:

```
stopDB.sh
```

Starting and Stopping the RulePoint Host Agent Instance

RulePoint requires the host agent instance to run on all individual hosts configured in a topology. Start the host agent to start all the required processes and collect the statistics. Each host agent communicates with the grid manager within the topology.

You can use the `startHostAgent` command to start the RulePoint host agent instance. The `startHostAgent` command has the following options:

```
startHostAgent
<-h> ip_address
<-help> help
<-p> port_number
```

The following table describes `startHostAgent` options and arguments:

Option	Argument	Description
-h	ip_address	IP address to bind the host agent.
-help	help	Usage of the <code>startHostAgent</code> command.
-p	port_number	Port number of the host agent service. The default port number is 19000.

You can use the `stopHostAgent` command to stop the RulePoint host agent instance. The `stopHostAgent` command has the following options:

```
stopHostAgent
<-h> ip_address
<-help> help
<-p> port_number
```

The following table describes the `stopHostAgent` options and arguments:

Option	Argument	Description
-h	ip_address	IP address of the host.
-help	help	Usage of the <code>stopHostAgent</code> command.
-p	port_number	Port number of the host agent service. The default port number is 19000.

Note: If you have registered Windows services for the RulePoint components during installation, you can stop or start the host agent instance from the `RulePoint HostAgent` service in Windows services.

Starting and Stopping the RulePoint Host Agent Instance on Windows

You can use the `startHostAgent.bat` command to start the RulePoint host agent instance on Windows.

RulePoint installs `startHostAgent.bat` in the following directory by default:

```
<RULEPOINT_HOME>/bin
```

1. Go to the directory where `startHostAgent.bat` is located.
2. At the command prompt, enter the following command to start the RulePoint host agent instance:

```
startHostAgent -h <ip_address> -p <port_number>
```

Enter the following command to stop the RulePoint host agent instance:

```
stopHostAgent -h <ip_address> -p <port_number>
```

Starting and Stopping the RulePoint Host Agent Instance on UNIX

You can use the `startHostAgent.sh` command to start the RulePoint host agent instance on UNIX.

RulePoint installs `startHostAgent.sh` in the following directory by default:

```
<RULEPOINT_HOME>/bin
```

1. Go to the directory where `startHostAgent.sh` is located.
2. At the command prompt, enter the following command to start the RulePoint host agent instance:

```
startHostAgent.sh -h <ip_address> -p <port_number>
```

Enter the following command to stop the RulePoint host agent instance:

```
stopHostAgent.sh -h <ip_address> -p <port_number>
```

Starting and Stopping RulePoint Design-Time Instance

The RulePoint design-time instance runs on a Tomcat web container. Use the design-time user interface to author objects, configure, and administer the RulePoint run time.

You need to consider the following tasks:

- After you start the host agent instance and topology instance, start the RulePoint design-time instance. You can manually start and stop the design-time instance.
- Clear the `\temp` directory.
- If you want to start the design-time instance of RulePoint on Suse Linux 10, make sure that the glibc version is 2.6 or later.

Note: If you have registered Windows services for the RulePoint components during installation, you can stop or start the design-time instance from the `RulePoint Design Time` service in Windows services.

Starting and Stopping RulePoint Design-Time Instance on Windows

You can use the `design.bat` command to start the RulePoint design-time instance on Windows.

RulePoint installs `design.bat` in the following directory by default:

```
<RULEPOINT_HOME>/bin
```

1. Go to the directory where `design.bat` is located.
2. At the command prompt, enter the following command to start the RulePoint design-time instance:

```
design start
```

Enter the following command to stop the RulePoint design-time instance:

```
design stop
```

Starting and Stopping RulePoint Design-Time Instance on UNIX

You can use the `design.sh` command to start the RulePoint design-time instance on UNIX.

RulePoint installs `design.sh` in the following directory by default:

```
<RULEPOINT_HOME>/bin
```

1. Go to the directory where `design.sh` is located.

2. At the command prompt, enter the following command to start the RulePoint design-time instance:

```
design.sh start
```

Enter the following command to stop the RulePoint design-time instance:

```
design.sh stop
```

Starting and Stopping RulePoint Topology Instance

RulePoint topology instance starts the run-time components of RulePoint. Use the RulePoint topology to deploy and process the design-time objects.

Note: If you have registered Windows services for the RulePoint components during installation, you can stop or start the topology instance from RulePoint Topology service in Windows services.

Complete the following tasks before you start the topology instance:

- Ensure that a valid license is available in the <RULEPOINT_HOME>\conf directory.
- Ensure that you clear the \temp directory.

Starting and Stopping RulePoint Topology Instance on Windows

You can use the topology.bat command to start the RulePoint topology instance on Windows.

RulePoint installs topology.bat in the following directory by default:

```
<RULEPOINT_HOME>/bin
```

1. Go to the directory where topology.bat is located.

The topology command has the following syntax:

```
topology start|shutdown <TopologyName>
```

2. At the command prompt, enter the following command to start the RulePoint topology instance:

```
topology start <TopologyName>
```

Enter the following command to stop the RulePoint topology instance:

```
topology shutdown <TopologyName>
```

Starting and Stopping RulePoint Topology Instance on UNIX

You can use the topology.sh command to start the RulePoint topology instance on UNIX.

RulePoint installs topology.sh in the following directory by default:

```
<RULEPOINT_HOME>/bin
```

1. Go to the directory where topology.sh is located.

The topology.sh command has the following syntax:

```
topology.sh start|shutdown <TopologyName>
```

2. At the command prompt, enter the following command to start the RulePoint topology instance:

```
topology.sh start <TopologyName>
```

Enter the following command to stop the RulePoint topology instance:

```
topology.sh shutdown <TopologyName>
```


Logging In to RulePoint

After you install RulePoint, use the default administrative account and password to access RulePoint. The default RulePoint administrator account user name is Administrator.

1. Launch a web browser.
2. In the address field, enter the following URL for the RulePoint login page:

`http://host:port/rulepoint`

The *host* represents the host name, the fully qualified domain name, or the IP address of the machine where you install RulePoint. The *port* is the HTTP port number of the Tomcat server. The default is 8080. You define the Tomcat port during installation.

The RulePoint login page appears.

3. Enter the user name and password.
4. Click **Log In**.

The RulePoint home page appears.

Configuring User Preferences

The User Preferences link is at the top right-hand corner of the page. On the User Information page, you can change your account information, including your name and password. You can also change the default date and time format for RulePoint sessions.

Changing Your Password

You must change the password the first time you log in to RulePoint. If you have a local user account, you can change your password.

1. In the RulePoint tool header area, click **User Preferences**.
2. Under **User Details**, click **Update**.
3. In the **Password** field, change the password.

The password must be 9 to 16 characters. It must contain only alphanumeric characters. It must have at least one alpha character and one numeric character.

4. Click **Save**.

Changing the Splash Screen Configuration

You can customize the RulePoint startup splash screen from the user preferences.

1. Click **User Preferences**.
The User Preferences dialog box appears.
2. Under the **Configuration** section, enable or disable the splash screen.

Changing the Date and Time Format

1. Click **User Preferences**.

The User Preferences dialog box appears.

2. Change the Date Format and the Time Format as required.

The default date format is DD/MM/YYYY and default time format is hh:mm:ss AM/PM.

The following table provides examples to show how the date and time patterns are interpreted:

Date and Time Pattern	Expected Result
"dddd, D MMMM, YYYY" at "HH:mm:ss"	Monday, 3 February, 2014 10:13:54
"MM-DD-YY" at "h:mm a"	02-03-14 10:13 am
"ddd, D MMMM, YYYY" at "HH:mm:ss"	Mon, 3 February, 2014 10:13:54
"MMMM D, YYYY" at "H:mm"	February 3, 2014 10:13

3. Click **Save**.

A message appears that prompts you to refresh your browser to apply the changes.

4. Click **OK**.

CHAPTER 3

Using RulePoint User Interface

This chapter includes the following topics:

- [Using RulePoint User Interface Overview, 27](#)
- [Administration Tab, 28](#)
- [User Management View, 28](#)
- [Import View, 29](#)
- [Export View, 30](#)
- [Topology View, 30](#)

Using RulePoint User Interface Overview

The user interface in RulePoint has a unified authoring environment, with integrated options to handle both the design-time and the run-time environment.

The user interface helps you build, deploy, and manage deployments. You have features to create RulePoint objects, configure the run-time components, deploy objects, and administer rule processing.

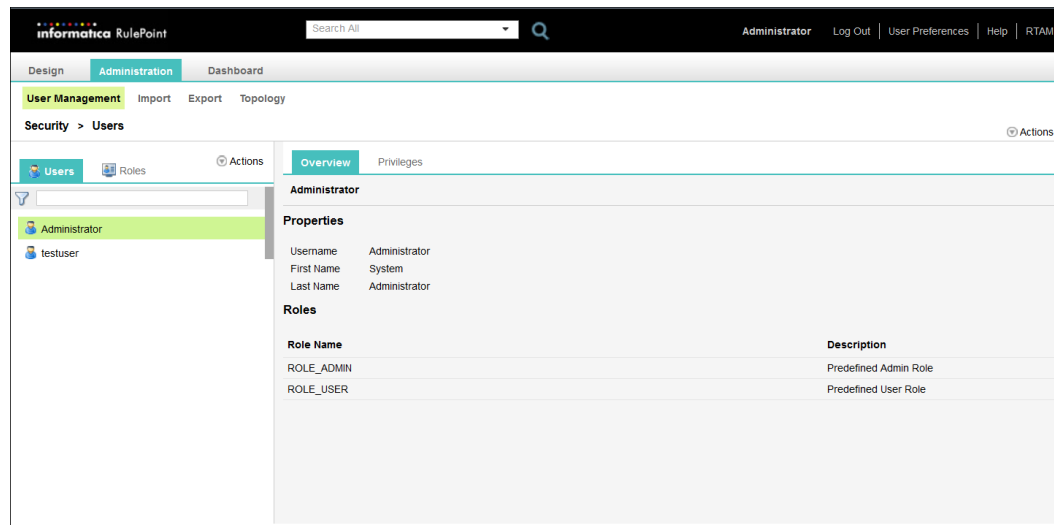
The user interface has the following tabs:

- Design. View and edit the properties of the project and objects within the project. For more information about the design tab, see the *RulePoint User Guide*.
- Administration. View user management, import and export objects, and configure a topology within a project.
- Dashboard. View the topology and deployed objects in a service, events, and logs.

The user interface has the following header items:

- Log out. Log out of the RulePoint user interface tool.
- User Preferences. Manage your account. The contents of the user information page that appears depends on whether your system has been configured to authenticate by using an external authentication provider. If your system is configured to authenticate by using the RulePoint database, this page enables you to update your user information.
- Help. Access help for the current tab and determine the RulePoint version.
- RTAM. Link to the RTAM dashboard.

The following image shows the **Administrator** tab of the RulePoint user interface:



Administration Tab

On the **Administration** tab, you can import and export objects, and manage users and topologies.

The contents that appear and the tasks that you can complete on the **Administration** tab depend on the view that you select.

Note: You can see the **Administration** tab only if you have the super user privileges.

You can select the following views from the **Administration** tab:

- User Management. Create, view, and manage all users and roles in a project.
- Import. View, upload, and import objects.
- Export. View and export objects.
- Topology. View, manage, and create the run-time components.

User Management View

The **User Management** view displays the users and the roles.

The contents that appear and the tasks you can complete on the **User Management** view vary based on the view that you select.

User View

The **User** view shows all the configured users and roles in the project.

The **User** view has the following components:

- Navigator. Appears in the left pane of the **User** view. The navigator displays the users in the project.

- Contents panel. Appears in the right pane of the **User** view and displays the overview of the properties and privileges for a user that you select in the navigator. The properties include the roles assigned and the associated privileges.
- **Actions** menu in the navigator. When you select the user in the navigator, you can create a user or delete a user.
- Action buttons in the **User Management > User** view. Appears in the right pane. When you select the user in the navigator, you can create, edit, or delete users. You can also configure LDAP for a user.

Role View

The **Role** view shows all the configured users and roles in the project.

The **Role** view has the following components:

- Navigator. Appears in the left pane of the **Role** view. The navigator displays the roles in a project.
- Contents panel. Appears in the right pane of the **Role** view and displays the overview of the properties and privileges for a particular role that you select in the navigator. The properties include the roles assigned to the user.
- **Actions** menu in the navigator. When you select the role in the navigator, you can create a role or delete a role.
- Action buttons in the **User Management > Role** view. Appears in the right pane. When you select the user in the navigator, you can create, edit, or delete roles.

Import View

You can import primary and secondary objects into RulePoint.

The **Import** view contains the following components:

- Navigator. Appears in the left pane of the **Import** view and displays the projects.
- **Actions** menu in the navigator. When you select a project in the navigator, you can upload a file into that project.
- Contents panel. Appears in the right pane of the **Import** view.
 - List View. In this view, you can view the import history and the available files in a project. When you select a file in the contents panel, use the menu displayed on its right to delete that file from a project.
 - Details View. In this view, you can view both the import history and the available files on the left pane, and the summary and details of the XML file on the right pane. Select the file in the **Import History** or **Available Files** tabs to view either the summary of each file, or the details, such as the object name, the object type, and status of each object within that XML file.
- **Actions** menu in the right panel. When you select a file, you can upload a file, or import a file to a project.

Export View

In the **Export** view, you can export objects from RulePoint.

The **Export** view contains the following components:

- **Navigator.** Appears in the left pane of the **Export** view and displays the projects.
- **Actions** menu in the navigator. When you select a project in the navigator, you can export all objects or export selected objects from a project.
- **Contents panel.** Appears in the right pane of the **Export** view.
 - **List View.** In this view, you can view the export history of the files in a project. You can view the file name and status of exported files in that project. When you select a file in the contents panel, use the menu displayed on its right to download the file to a location that you can specify, or to delete the file from that project.
 - **Details View.** In this view, you can view the export history in that project on the left pane. Select a file to view either the summary of each exported file, or the details, such as the object name, the object type, and export status of that object within the XML file.
- **Actions** menu in the right panel. When you select a project, you can export all objects or export selected objects from a project.

Topology View

In the **Topology** view, you can view information about the host, node, application services, and the system services for all configured topologies. You can view, configure, and manage the hosts, nodes, and services in a topology.

The **Topology** view displays the following information:

- **Navigator.** Appears in the left panel of the **Topology** view and displays the name of the topology and the components in the topology.
- **List View.** Appears in the right panel of the **Topology** view. When you select the topology in the navigator, you can view the name of the topology and its last modified details. You can also edit or add a host to a topology from the menu on the right.
- **Details View.** When you select the topology in the left panel, the details view displays the name, description, created and modified details for the topology. When you select the properties view, you can view the configured values for the topology.

Host

A host is a physical machine in the topology. In the **Topology** view, you assign nodes, UM store and UM lbmrd to a host.

When you select a host in the navigator, the components of the user interface display the following information about the host and let you complete tasks for the host:

- **List View.** Appears in the right panel of the **Topology** view. You can view the host name, port number, and last modified date. You can also edit the host, add a node, or add a UM store to the host from the menu on the right. You can also delete a host.

- Details View. When you select the host in the left panel, the details view displays the name, description, port number, created and modified details for the host. When you select the properties view, you can view the configured values for the host.
- Action menu in the right panel. You can add a host to the topology.

Node

A node runs the service controller, event processor, responder controller, and the activity manager.

When you select a node in the navigator, the components of the user interface display the following information about the node and let you complete tasks for the node:

- List View. Appears in the right panel of the **Topology** view. You can view the node name, host name, port number, and last modified date. You can also edit or delete the node. You can also add a source controller, event processor, responder controller, or activity manager from the menu on the right.
- Details View. When you select the node in the left panel, the details view displays the name, description, host IP address, port number, created and modified details of the node. When you select the properties view, you can view the configured values for the node.
- Action menu in the right panel. You can add a node to the host.

Application Services

Application services comprise the source controller, responder controller, event processor, and activity manager.

Source Controller

A source controller runs on a node. A source controller receives inbound events from a configured source.

When you select a source controller in the navigator, the components of the user interface display the following information and let you complete tasks for the source controller:

- List View. Appears in the right panel of the **Topology** view. You can view the name of the source controller, node name, and last modified date. You can also edit or delete a source controller from the menu on the right.
- Details View. When you select the source controller in the left panel, the details view displays the name, description of the source controller, created and modified details for the source controller. When you select the properties view, you can view the configured values for the source controller.
- Actions menu in the right panel. You can add a source controller on standalone mode or high availability mode to a node.

Event Processor

An event processor runs on a node. A event processor processes the events against the configured rules.

When you select an event processor in the navigator, the components of the user interface display the following information and let you complete tasks for the event processor:

- List View. Appears in the right panel of the **Topology** view. You can view the name of the event processor, node name, and last modified date. You can also edit or delete a event processor from the menu on the right.

- **Details View.** When you select the event processor in the left panel, the details view displays the name, description of the event processor, created and modified details for the host. When you select the properties view, you can view the configured values for the event processor.
- **Actions menu in the right panel.** You can add a event processor on standalone mode or high availability mode to a node.

Responder Controller

A responder controller runs on a node. A responder controller transmits events from a configured source.

When you select a source controller in the navigator, the components of the user interface display the following information and let you complete tasks for the responder controller:

- **List View.** Appears in the right panel of the **Topology** view. You can view the name of the responder controller, node name, and last modified date. You can also edit or delete a responder controller from the menu on the right.
- **Details View.** When you select the responder controller in the left panel, the details view displays the name, description of the responder controller, created and modified details for the responder controller. When you select the properties view, you can view the configured values for the responder controller.
- **Actions menu in the right panel.** You can add a responder controller on standalone mode or high availability mode to a node.

Activity Manager

The activity manager runs on a node. An activity manager records all the activity in the system.

When you select the activity manager in the navigator, the components of the user interface display the following information and let you complete tasks for the activity manager:

- **List View.** Appears in the right panel of the **Topology** view. You can view the name of the activity manager, node name, and last modified date. You can also edit or delete an activity manager from the menu on the right.
- **Details View.** When you select the activity manager in the left panel, the details view displays the name, description of the activity manager, created and modified details for the activity manager. When you select the properties view, you can view the configured values for the activity manager.
- **Actions menu in the right panel.** You can add a activity manager on standalone mode or high availability mode to a node.

System Services

System services comprise the grid manager, UM store, and UM lbmrd.

Grid Manager

The grid manager runs on a host, and manages all the run-time components.

When you select a grid manager in the navigator, the components of the user interface display the following information and let you complete tasks for the grid manager:

- **List View.** Appears in the right panel of the **Topology** view. You can view the name of the grid manager, host IP address, port number, and last modified date. You can also edit or delete a grid manager from the menu on the right.

- Details View. When you select the grid manager in the left panel, the details view displays the name, description, host IP address, port number, created and modified details of the grid manager. When you select the properties view, you can view the configured values for the grid manager.
- Action menu in the right panel. You can add a grid manager to the topology.

UM Store

The UM store is the messaging layer that assists communication of the run-time components.

When you select a UM store in the navigator, the components of the user interface display the following information and let you complete tasks for the UM store:

- List View. Appears in the right panel of the **Topology** view. You can view the name of the UM store, host IP address, port number, and last modified date. You can also edit or delete a UM store from the menu on the right.
- Details View. When you select the UM store in the left panel, the details view displays the name, description, host IP address, port number, created and modified details of the UM store. When you select the properties view, you can view the configured values for the UM store.
- Action menu in the right panel. You can add a UM store to the topology.

UM lbmrd

The UM lbmrd facilitates communication of the run-time components.

When you select a UM lbmrd in the navigator, the components of the user interface display the following information and let you complete tasks for the UM lbmrd:

- List View. Appears in the right panel of the **Topology** view. You can view the name of the UM lbmrd, host IP address, port number, and last modified date. You can also edit or delete a UM lbmrd from the menu on the right.
- Details View. When you select the UM lbmrd in the left panel, the details view displays the name, description, host IP address, port number, created and modified details of the UM lbmrd. When you select the properties view, you can view the configured values for the UM lbmrd.
- Action menu in the right panel. You can add a UM lbmrd to the topology.

CHAPTER 4

User Management

This chapter includes the following topics:

- [User Management Overview, 34](#)
- [Access Control Lists in RulePoint, 35](#)
- [Authentication, 37](#)
- [Best Practices for User Management, 45](#)

User Management Overview

RulePoint includes an administrator user and default roles and privileges. If you have the super user privileges, you can create users and assign roles to users.

You can assign privileges to user roles. You can associate a user with one or more roles. A user acquires all privileges attached to assigned roles. You can customize a role by associating a set of privileges to roles.

You can configure Access Control List (ACL) to manage access to RulePoint objects. You can edit direct permissions from the ACL list to grant or deny object permissions to a user. For example, you can define if a user can update or delete an instance of a source. You can add permissions to a local user, role, LDAP user, or LDAP group.

Default Users, Roles, and Privileges

The administrator is the default user created during RulePoint installation. Role_User and Role_Admin are the default roles created for the administrator.

As a Role_User and Role_Admin, you have a set of predefined view and create privileges that you can administer for rules, watchlists, topics, analytics, connection, project, source, template, response, and responder. In addition to the listed privileges, the Role_Admin also includes super user privileges. Only a user with the super user privileges can perform the user management tasks.

As an administrator user, you can access all RulePoint objects, and can create and manage other users in RulePoint. You can assign roles, permissions, and privileges to users, which determine what tasks users can perform within RulePoint. You can assign a user with more than one role, based on the requirement. You can delete users and roles that you create. You cannot delete the administrator user or system-defined roles.

Privilege Types for RulePoint Users

In RulePoint, you have privileges defined for each object type and project. You can assign privileges to roles based on what function you want the user to perform. You assign privileges to users through roles. Users inherit all the privileges associated with the roles.

User roles include all the listed privileges, while administrator roles, additionally, include the PRIV_SUPER_USER. An administrator with the super user privileges can perform all administration operations, such as user management, import, export, and topology functions.

All privileges are independent. For example, you have a privilege to create and view sources, but do not have the privileges to view topics. When you attempt to configure a topic for a source, a message appears, which states that you do not have the privileges to view the topic.

The following table lists the privileges associated with RulePoint objects:

Object	Associated Privileges
Rule	<ul style="list-style-type: none">- PRIV_VIEW_RULE- PRIV_CREATE_RULE
Watchlist	<ul style="list-style-type: none">- PRIV_VIEW_WATCHLIST- PRIV_CREATE_WATCHLIST
Topic	<ul style="list-style-type: none">- PRIV_VIEW_TOPIC- PRIV_CREATE_TOPIC
Analytic	<ul style="list-style-type: none">- PRIV_VIEW_ANALYTIC- PRIV_CREATE_ANALYTIC
Connection	<ul style="list-style-type: none">- PRIV_VIEW_CONNECTION- PRIV_CREATE_CONNECTION
Project	<ul style="list-style-type: none">- PRIV_VIEW_PROJECT- PRIV_CREATE_PROJECT
Source	<ul style="list-style-type: none">- PRIV_VIEW_SOURCE- PRIV_CREATE_SOURCE
Template	<ul style="list-style-type: none">- PRIV_VIEW_TEMPLATE- PRIV_CREATE_TEMPLATE
Response	<ul style="list-style-type: none">- PRIV_VIEW_RESPONSE- PRIV_CREATE_RESPONSE
Responder	<ul style="list-style-type: none">- PRIV_VIEW_RESPONDER- PRIV_CREATE_RESPONDER

Access Control Lists in RulePoint

Each instance of the object or project has an associated ACL object.

Each access control entry in an ACL object consists of the following information:

- User name

- Permission, such as read, write, execute, and admin
- Action, such as grant or deny access

You can edit permissions or add permissions to a local or LDAP user or role. For information about ACLs and configuring access controls, see the *RulePoint User Guide*.

Permission Types

Permissions in RulePoint are independent. If you have a privilege to create a specific object, you do not inherit the privilege to view that object. If you need all the available permissions, you need to include all those permissions explicitly. You can grant or deny permissions for user access to objects.

The following table lists the actions for each type of permission:

Permission Type	Action
Read	View an instance of the object. When you have read permission, you can perform tasks in which you view details associated with the object.
Write	Edit and delete an object.
Execute	Deploy, undeploy, redeploy, and reassign a primary object. Use an execute permission to perform the deploy-related actions on the object.
Admin	Edit the ACL of an object, for example, add or remove entries.

ACL Permission Rules

You can control RulePoint user or role access through ACL rules. The ACL rules require you to pass a set of requirements to gain access to a particular object.

In RulePoint, consider the following ACL rules:

- If a user has ACL permissions for a project, that user inherits the ACL permissions for all the objects within that project, unless there is a more specific ACL permission entry at the object level that overrides it.
- If a role has ACL permissions for a project, the user associated with that role inherits the ACL permissions for all the objects within that project, unless there is a more specific ACL permission entry at the object level that overrides it.
- When a user creates an instance of an object, the ACL rule created for that object includes all the four permissions for that user, such as read, write, execute, and admin.
- If a user creates an instance of the object, that user cannot modify the ACL settings in it.
- Users with the super user privilege can access all objects irrespective of the ACL entry.

Evaluating ACL Rule Permission Requirements

ACL rules determine user access to an object if the user meets all permissions required by the matching ACL rules. When a user attempts to access an object, RulePoint evaluates permissions of the matching ACL rules.

Based on the evaluation, RulePoint either grants or denies user access based on the following matching ACL rules:

- If the ACL has a matching entry, where the user ID is the logged in user, has the required permissions, and the grant access for the permissions, the user can access the RulePoint object. For example, if a user has write permission for a specific object with grant access, that user can create, edit, or delete that object.
- If the ACL has a matching entry, where the user ID is the logged in user, has the required permissions, but the deny access for the permissions, RulePoint denies user access to the object. For example, if a user has write permission for a specific object with deny access, that user cannot edit or delete that object.
- If the ACL does not have a matching entry according to the first two rules, RulePoint invokes the parent ACL and runs the rules again until it finds a matching entry. When RulePoint reaches the end of the ACL hierarchy, and there are no more parent ACLs, it denies user access to the object. For example, if a user has read and write permission for a specific project, but no grant or deny access specified at the object level, that user will have the read and write access on all the objects in that project.

Authentication

When you log in to RulePoint, RulePoint authenticates your credentials by using the local or remote authentication provider. User access is further based on the configured access control list for each object and the groups to which you have assigned the user in the authentication provider.

You can use more than one type of authentication in RulePoint. Consider the following options for configuring authentication in RulePoint:

Local authentication in RulePoint database

When you log in to RulePoint as a local user, RulePoint authenticates your user name and password against the user accounts.

Remote authentication using Lightweight Directory Access Protocol (LDAP) Server

When you log in to RulePoint as an LDAP user, RulePoint passes your user name and password to the external directory service for authentication. The directory service can be Open LDAP Directory Service or the Microsoft Active Directory.

By default, RulePoint uses local authentication. If you use the default RulePoint database for authentication, two roles, Role_User and Role_Admin, are available. For creating additional user roles in your implementation, you must use either a local authentication server or a remote authentication provider. If you use an external authentication provider, RulePoint will be as secure as the external authentication server. Verify that you secure the authentication server.

LDAP Authentication

For user accounts in an enterprise LDAP directory service to have RulePoint access, you need to configure RulePoint to use LDAP authentication.

Configuring LDAP for RulePoint Users

If you configure RulePoint to use LDAP, you must configure LDAP from the **Administrator** tab in the user interface. Only an administrator can configure the LDAP settings.

1. On the **Administrator** tab, click the **User Management** view.
2. In the navigator, click the **Users** or the **Roles** view.
3. From the **Actions** menu in the upper-right panel, select **LDAP Configuration**.
4. In the **LDAP Configuration- Step 1 of 2** page, click **Enable LDAP Configuration**.
5. Under **LDAP Server Details**, configure the LDAP server properties.

The following table describes the LDAP server configuration properties:

Property	Description
Encryption Type	Defines the LDAP transport security definition. Choose No Encryption or SSL Encryption . To connect to the LDAP server with SSL encryption, you must import the generated CA certificate from the supported LDAP server into the java keystore.
Server Name	Name of the machine that hosts the LDAP directory service, for example, Open LDAP Directory Service or the Microsoft Active Directory.
Port	The listening port number of the LDAP server. The default LDAP server port number is 389.

6. Under **Principal User Details**, configure the principal LDAP user properties.

The following table describes the LDAP user properties:

Property	Description
Distinguished Name	The name of the principal user, who is an administrative user with access to the LDAP directory. The user name often consists of a common name (CN), an organization (O), and a country (C). The principal user name is an administrative user with access to the directory. Specify a user that has permission to read other user entries in the LDAP directory service. Leave blank for anonymous login. For more information, see the documentation for the LDAP directory service.
Password	The password for the principal user that RulePoint uses to connect to LDAP. The name must not start with numbers or whitespace.
Group Membership Attribute	The name of the attribute that contains group membership information for a user. This is the attribute in the LDAP group object that contains the DNs of the users or groups who are members of a group. For example, member or memberof. The default LDAP attribute is uniqueMember.

7. Under **Group Membership Attribute**, enter the name of the attribute that contains group membership information for a user.

This is the attribute in the LDAP group object that contains the DNs of the users or groups who are members of a group. For example, member or memberof. The default LDAP attribute is uniqueMember.

8. Click **Test Connection** to verify that the connection configuration is correct.
9. Click **Next**.

The **LDAP Configuration - Step 2 of 2** dialog box appears.

10. Under **LDAP Setup > Search Criteria**, provide the search criteria to filter users and groups that you want to include in this security domain.

The following table describes the search criteria that you can set up to filter users and groups in the LDAP directory service to access RulePoint:

Property	Description
User Search Base	Distinguished name (DN) of the entry that serves as the starting point to search for user names in the LDAP directory service. The search finds an object in the directory according to the path in the distinguished name of the object. For example, in Microsoft Active Directory, the distinguished name of a user object might be <code>cn=UserName,ou=OrganizationalUnit,dc=DomainName</code> , where the series of relative distinguished names denoted by <code>dc=DomainName</code> identifies the DNS domain of the object.
User DN Pattern	A DN pattern used to directly log in users to the LDAP database. This pattern is used for creating a DN string for direct user authentication, where the pattern is relative to the base DN in the LDAP URL. For example: <code>cn={0},ou=People</code>
Group Search Base	Distinguished name (DN) of the entry that serves as the starting point to search for group names in the LDAP directory service.
User Filter	An LDAP query string that specifies the criteria for searching for users in the directory service. The filter can specify attribute types, assertion values, and matching criteria. For example: <code>(objectclass=*)</code> searches all objects. <code>(&(objectClass=user)(!(cn=susan)))</code> searches all user objects except "susan." For more information about search filters, see the documentation for the LDAP directory service. The default value is <code>objectClass=person</code> .
Group Filter	The name of the LDAP query string that specifies the criteria for searching groups in the directory service. The default value is <code>objectClass=groupofuniquenames</code> .
User Search Attribute	The LDAP attribute that you want RulePoint to use when searching for a user in the user interface. The default value is relative to the security provider. For example, the default for Active Directory is <code>uid={0}</code> and for Open LDAP the default is <code>UserPrincipalName</code> .
User Search Value Format	The string that RulePoint must pass to the authentication server when authenticating a user, and substitutes {0} with the username supplied by the end user. For example, in LDAP: <code>securityProvider.ldapUserDnFormat=uid={0},ou=employees,dc=dev,dc=agentlogic,dc=com</code> For example, in AD: <code>securityProvider.ldapUserDnFormat={0}@company.com</code> where <code>{0}@company.com</code> represents the format of the user's <code>userPrincipalName</code> as set in your AD server.
Group Display Name Attribute	The LDAP attribute that you want RulePoint to display for a group in the user interface. Although you can choose to display any LDAP attribute, <code>cn</code> is the recommended value. The default value is <code>cn</code> .
User Display Name Attribute	The LDAP attribute that you want RulePoint to display for a user in the user interface. Although you can choose to display any LDAP attribute, <code>cn</code> is the recommended value. The default value is <code>cn</code> .

Property	Description
First Name Attribute	The LDAP attribute that contains each user's given name. The default value is givenName.
User Last Name Attribute	The LDAP attribute that contains each user's last name or family name. The default value is sn.

11. Click **Preview Search Criteria** to view the list of users and groups that fall within the filter parameters.
12. Click **Save**.
The configured settings for LDAP are saved in the system.

Disabling LDAP Configuration

If you do not want to provide access to LDAP users, you can disable LDAP. After you disable, LDAP users cannot log in to RulePoint.

1. On the **Administrator** tab, click the **User Management** view.
2. From the **Actions** menu in the upper-right panel, select **LDAP Configuration**.
3. Under **LDAP Configuration**, disable the LDAP configuration, and click **Next**.
4. Click **Save**.

Configuring LDAP Authentication for RTAM

When users log in to RTAM, the user credentials are authenticated using the authentication provider (whether local or external) and mechanism configured for RulePoint. Every user authenticated is granted full access to the application.

By default, RTAM authenticates users using the RulePoint database as the provider. If you have configured LDAP for RulePoint, you need to configure LDAP authentication for RTAM.

Changing the Default Authentication Provider to LDAP

You need to change the default authentication to LDAP:

- To configure RTAM to use LDAP, edit the contextConfigLocation property in the `web.xml` file located in the `[RTAMHome]\WEB-INF\` directory to include the following information:

```
<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>
/WEB-INF/classes/spring-context.xml
/WEB-INF/classes/security-ldap.xml
</param-value>
</context-param>
```


Configuring the Authentication Provider

After selecting the authentication provider that you want to use, you must edit the `rtamconfig.properties` file to identify the location and the relevant login credentials of the selected authentication provider.

1. To configure RTAM to use RulePoint for authentication, edit the following RulePoint internal authentication setting in the `rtamconfig.properties` file to match the login credentials provided by RulePoint to its security provider:

```
ldap.provider.url=  
ldap.user.search.base=  
ldap.user.search.filter=sAMAccountName={0}  
ldap.group.search.base=  
ldap.role.default=ROLE_USER  
ldap.role.search.subtree=true  
ldap.role.prefix=ROLE_EXT_  
ldap.role.convert.touppercase=true  
ldap.user.default=  
ldap.password.default=  
ldap.group.search.filter=  
ldap.group.role.attribute=memberOf  
ldap.username.format={0}@
```

Note: You might not require some of the listed properties to connect to your authentication provider. For specific configuration settings, contact your LDAP administrator for information.

2. If you use open LDAP for authentication, use the following sample configuration properties:

```
ldap.provider.url=ldap://  
ldap.user.search.base=  
ldap.user.search.filter=(uid={0})  
ldap.group.search.base=  
ldap.group.search.filter=(&(objectClass=groupofuniqueNames)  
(uniqueMember={0}))  
ldap.group.role.attribute=  
ldap.role.default=ROLE_USER  
ldap.role.search.subtree=true  
ldap.role.prefix=ROLE_EXT_  
ldap.role.convert.touppercase=true  
ldap.username.format={0}@  
ldap.user.default=  
ldap.password.default=
```

Note: You might not require some of the listed properties to connect to your authentication provider. For your specific configuration settings, contact your LDAP administrator for information.

3. If the LDAP password is encrypted, you must add `<value>ldap.password.default</value>` for `encryptedPropertyNames` in the `spring-context.xml` file at `<RULEPOINT_HOME>\design\webapps\RTAM\WEB-INF\classes`:

```
</property>  
<property name="encryptedPropertyNames">  
  <set value-type="java.lang.String">  
    <value>hibernate.password</value>  
    <value>auth.db.password</value>  
    <value>ldap.password.default</value>  
  </set>  
</property>
```

Configuring the Group Resolver

RTAM supports group addresses for both standard LDAP and open LDAP. The group resolver determines and resolves the individual names within the given group.

1. If you use the standard LDAP group resolver, use the following properties:

```
ldapgroupresolver.url=ldap://ldapserver.mycompany.com:389  
ldapgroupresolver.base=CN=Users,DC=mycompany,DC=com
```

```

ldapgroupresolver.userDn=user1@mycompany.com
ldapgroupresolver.userPwd=xxxxxxx
ldapgroupresolver.baseFilter=
ldapgroupresolver.filter=(&(objectclass=person)
(memberOf=CN=%s,OU=Mailing List Groups,DC=mycompany,DC=com))
ldapgroupresolver.uidAttribute=sAMAccountName
ldapgroupresolver.escapeGroupIds=false
ldapgroupresolver.maxAge=3600000

```

2. If you use the open LDAP group resolver, use the following properties:

```

ldapgroupresolver.url=ldap://
ldapgroupresolver.base=
ldapgroupresolver.userDn=
ldapgroupresolver.userPwd=
ldapgroupresolver.baseFilter=
ldapgroupresolver.filter=(&(objectclass=person) (ou:dn=%s))
ldapgroupresolver.uidAttribute=uid
ldapgroupresolver.escapeGroupIds=false

```

Managing Users and Roles

You can create, edit, and delete users.

You can create roles and assign roles to users. The roles and privileges assigned to the user determine the tasks that the user can perform within RulePoint. You can also delete a role.

Creating Local Users

You can create users in RulePoint. The roles, permissions, and privileges that you assign to a user determine the tasks that the user can perform within RulePoint. When you create a user, you need to specify the authentication provider as local.

1. On the **Administration** tab, click **User Management > Users** view.
2. Select **New User** from the **Actions** menu in the navigator or in the contents pane.
3. Under **Details**, select **Type** as **Local**, and provide the following configuration details:
 - a. Under **Username**, provide a name for the user.
 - b. Under **First Name**, provide the first name of the user.
 - c. Under **Password**, provide the password for the user.
The password must be between 9 and 16 characters. It must contain only alphanumeric characters. It must have at least one alpha character and one numeric character.
 - d. Under **Last Name**, provide the last name of the user.
4. Click **Next**, and perform one of the following tasks:
 - Click **Select All** to select all the available roles.
 - Select one or more roles for the user from the list displayed.
5. Click **Save**.

Creating Remote Users

You can create remote users in RulePoint. The roles, permissions, and privileges that you assign to a user determine the tasks that the user can perform within RulePoint. When you create a remote user, you need to specify the authentication provider as remote.

Before you create a remote user, you must configure LDAP.

1. On the **Administration** tab, click **User Management > Users** view.

2. Select **New User** from the **Actions** menu in the navigator or in the contents pane.
3. Under **Details**, select **Remote**.
4. Under **Remote User DN**, type the user distinguished name. You can search for the remote user DN from the LDAP directory.
Under **Search Results**, the display name and user DN appear.
5. Under **Username**, provide the user name for the remote user. You can use the same name as the LDAP user.
6. Click **Next**.
7. Select one or more roles for the user from the list displayed, and click **Save**.

Editing a User

You can edit properties of a user in RulePoint. You can reassign roles, permissions, and privileges to a user.

1. On the **Administration** tab, click **User Management > Users** view.
2. In the navigator, select the user you want to delete.
3. Select **Edit User** from the **Actions** menu on the right pane.
4. Under **Details**, select **Type** as local or remote, and edit the configuration details.
You cannot change the authentication type from local user to LDAP user, or LDAP user to local user.
5. Click **Next**.
6. Select one or more roles for the user from the list displayed, and click **Save**.

Deleting a User

You cannot delete the default user created during installation time.

1. On the **Administration** tab, click **User Management > Users** view.
2. In the navigator, select the user that you want to delete.
3. Select **Delete User** from the **Actions** menu in the navigator or in the contents pane.
4. If you have no ACLs configured for the user, a message appears that prompts you if you want to delete the user. Click **OK**.
5. If you configured ACLs for the local or LDAP user, a message appears, indicating that ACLs are configured for the user other than the default entries, and prompts you to first transfer the ACLs to a user or role and then delete the user. Perform the following tasks:
 - a. To first transfer the ACLs, and then delete the user, click **OK**.
 - b. In the **Transfer User ACL** dialog box, search for the user or role to whom you want to transfer the ACLs.
 - c. Select the user so that the user name displays on the **Transfer ACL To User** field, and click **OK**.
6. If you do not want to transfer the ACLs, click **Force Delete**, and then click **OK** when a confirmation message appears that the users are deleted successfully.

Creating a Role for a Local User

You can create roles for local users in RulePoint. You can associate one or more roles with a user. You must map privileges to the roles based on the type of operation you want the user to perform.

1. On the **Administration** tab, click **User Management > Roles** view.

2. Select **New Role** from the **Actions** menu in the contents pane.
3. Under **Name**, type a name for the role.
4. Under **Description**, type a description for the role.
5. Click **Next**.
6. Under **Role**, select the privileges you want to associate to the role, or use **Select All** on the upper-right pane to select all the privileges.
7. Click **Save**.

Creating a Role for a Remote User

You can create roles for remote users in RulePoint. If you create a role for a group of remote users, all users within that group will have access to RulePoint.

1. On the **Administration** tab, click **User Management > Roles** view.
2. Select **New Role** from the **Actions** menu in the navigator or in the contents pane.
3. Under **Name**, type a name for the role.
4. Under **Description**, type a description for the role.
5. Under **LDAP Group DNs**, type the distinguished name of the group to fetch a list of users in the LDAP directory service that you want to associate to the role, and gain access to RulePoint.
6. Click **Add** to add more DNs, and click **Next**.
7. Under **Role**, select the privileges you want to associate to the role, or use **Select All** on the upper-right pane to select all the privileges.
8. Click **Save**.

Deleting a Role

You need to transfer the ACLs associated to the roles before you delete the role.

1. On the **Administration** tab, click **User Management > Roles** view.
2. In the navigator, select the role you want to delete.
3. Select **Delete Role** from the **Actions** menu in the navigator or in the contents pane.
4. If you have no ACLs configured for the role, a message appears that prompts you if you want to delete the role. Click **OK**.
5. If you configured ACLs for the role associated to the local or LDAP role, a message appears which states that ACLs are configured for the role, and prompts you to first transfer the ACLs to another user or role. Perform the following tasks:
 - a. Click **OK**.
 - b. Search and select the role or user to which you want to transfer the ACLs, and then click **OK**.
6. If you configured ACLs for the LDAP group users associated to a role, and you want to transfer the ACLs to a user or role, perform the following steps:
 - a. Click **OK**.
 - b. To transfer the ACLs to the local user, select **Default User**, and select the user from the list generated from the search, and click **OK** to assign all the ACLs to the default user.
 - c. To transfer the ACLs to a group of users, select **Map Each User**. Select the users from the list generated from the search, assign ACLs to specific users, and then click **OK**.

- d. Select the roles from the generated list, and click **OK** to assign the ACLs to the selected roles.
7. If you configured ACLs for an LDAP role and a user who belongs to that LDAP group and you want to transfer the ACLs, perform the following steps:
 - a. To transfer the ACLs to a role, search for the user or role to which you want to transfer the ACLs, select the roles or users, and click **OK**.
 - b. To transfer the ACLs to the user, select **Default User**, and select the user from the list generated from the search, and click **OK** to assign all the ACLs to the default user.
 - c. To transfer the ACLs to a group of users, select **Map Each User**. Select the users or roles from the list generated from the search, assign ACLs to specific users or roles, and then click **OK**.
8. If you do not want to transfer the ACLs, click **Force Delete**, and then click **OK** when a confirmation message appears that the roles are deleted successfully.

Best Practices for User Management

Consider the following best practices when you create users or roles:

- Distribute permissions across multiple users or roles. For example, distribute permissions to create rules, sources, and responders.
- You can restrict permissions by using the ACLs at the object level, or you can apply ACLs at the project level.
- You can provide permissions for users or roles at the object level. For example, you can configure permission for a particular user or role to create a set of sources.

CHAPTER 5

Topology Management

This chapter includes the following topics:

- [Topology Management Overview, 46](#)
- [Topology Planning, 46](#)
- [Topology Design, 47](#)
- [Topology Design Considerations, 48](#)
- [Configuring a Topology, 49](#)

Topology Management Overview

The RulePoint topology is the physical layout of the run-time environment. A topology is a collection of hosts, nodes, and services that define the run-time environment.

The run-time architecture in RulePoint is flexible, and you can create a variety of highly available and scalable design patterns to improve the RulePoint performance. The default topology is the starting point, where you can build multiple application and system services. You can configure the application services on multiple nodes distributed on multiple host machines. You can also choose the deployment mode for application services as standalone or high availability.

The topology can consist of numerous source controllers, event processors, and responder controllers that you can stack together in a logical order to distribute your application objects. A topology can consist of many topology units and each unit might contain a source controller, responder controller and event processor. You can also add more CPUs and memory to the host if you want to increase the capacity and performance of each topology unit.

Designing a good topology for a highly scalable environment requires an appropriate strategy for mapping application load requirements to the logical topology units. You need to consider the amount of events, rules, potential alerts, environmental constraints, and performance goals.

Topology Planning

Use the **Topology** view on the **Administration** tab to create a custom topology environment that suits your application event processing requirements.

When you plan to expand a default topology, you must understand the complexity of managing and configuring the run-time components for effective performance, high availability, and scalability. It involves

choosing the right number of nodes, hosts, application services, and system services to maximize RulePoint performance. You also need to consider the overall application design, event volume, and rule volume.

Before you build your topology, you need to consider the following requirements for your run-time environment:

- Identify the capacity and performance requirements for your environment.
- Identify the functional requirements of the run-time environment, whether you want to configure the run-time components for scalability, high availability, or both.
- Identify the deployment requirements for the application. For example, if you configure the run-time environment for scalability, you must consider the following factors:
 - Understand the sizing and partitioning requirements for events and rules across corresponding application services.
 - Decide what objects you want to map to corresponding application services in the run-time environment.
- Configure the run-time environment by using the **Topology** view from the **Administration** tab. To choose a topology design:
 - Identify the number of nodes and hosts you need.
 - Identify the number of application services and system services you need.
 - Identify the object types that you will deploy for a particular application service instance, and the set of application services designated to interact together to process those objects.
 - Identify the objects for import or export into specific projects.
- Understand the tasks to configure the components of the run-time environment. For more information to configure additional nodes, hosts, application services, and system services, see specific instructions in this guide.
- Understand the best practices for building the run-time architecture.

Topology Design

You can configure the following topology designs:

- You can vertically scale the services by adding more CPUs and memory to the host. You can increase the heap size of a node and the threadpool size of the application services to take advantage of the additional CPU and memory.
- You can horizontally scale the topology by configuring multiple instances of source controllers, event processors, and responder controllers. Each node might contain one or more of the application services. The application services on a node can be of similar type or mixed, as required. You can partition sources, rules, and responders across corresponding application services across multiple nodes.
- When you configure the run-time components for scaling, you can partition the service workload of the sources, rules, and responders, across the services. You can dedicate specific nodes for custom services, heavy load sources, and high-throughput rules.
- You can configure a high-availability solution to eliminate single points of failure by replicating the application services and the system services as primary and backup instances. You can distribute these services across nodes to ensure that even if one node goes down, the backup instances are still available and continue to generate events, process rules, and dispatch alerts.

Topology Design Considerations

The scenarios provide various service combinations that you can refer to when you build a topology. Customize the topology based on your requirements.

Scenario 1: You want to create a development or testing environment to handle objects.

You can use the default topology created during RulePoint installation, which consists of a single source controller, event processor, responder controller, and supporting system services. The default topology setup is sufficient for development and unit testing purposes.

With increased production requirements, you can create more application services and partition objects to run on different services based on the application scalability and overall system stability needs.

Scenario 2: Your organization handles voluminous data (events, rules, and alerts) and you want to scale your application.

You can consider using the default topology, which consists of one source controller, event processor, and responder controller that run on separate nodes as a single topology unit, with a certain measured processing capacity.

If the overall application data load is more than what a single topology unit can handle, you need to build a topology that consists of multiple topology units, and partition the deployment of application objects across corresponding topology units.

Scenario 3: You have large number of rules to process same event input volume.

When you want to add many rules to process an event, the load on an event processor might increase and might impact the performance. To increase the processing capacity of events and the overall throughput, add an additional event processor. You must configure the event processors to run on two separate nodes, and partition the deployment of rules across both the event processors.

You can also reassign rules to different event processors when the load of evaluations for a particular event increases.

Scenario 4: You want to eliminate single points of failure for selective or all application services.

You can configure application services within the topology to run in high availability mode. You can create backups for the source controller, event processor, responder controller, and activity manager. The primary and backup instances must run on two separate nodes. Each set of primary and backup instance performs a specific function. When the primary instance fails, the backup instance becomes the primary instance and takes on the active role. You can also have multiple backups for each primary instance. When the primary instance fails, the grid manager elects one of the backup instances as the primary, which then takes on the role of the primary.

If you configure the nodes to run on two separate hosts, if one host fails, the nodes on the other host continue to work, eliminating any possibility of failure.

Scenario 5: You have custom sources and responders that are thread unsafe, and you want to isolate other objects from any failure or process crashes.

If you have custom sources and responders that are thread unsafe, it might bring down the node it is running in. For fault tolerance, it is important to configure a separate source controller and a responder controller and map them to nodes dedicated only for these services. You can then deploy the custom sources and responders on these nodes to safeguard performance of other objects in the system.

Scenario 6: You want to protect system services against single points of failure.

You can create multiple grid managers. Each host can have a single instance of the grid manager. If one grid manager on one host machine fails, the other instance takes on the activity.

You can also configure multiple instances of the UM store and lbmrd in a fault-tolerant manner to protect against individual store failure. All the configured persistent stores retain messages for recovery. If one of the persistent stores fails, the application services can use the other persistent store to recover and read messages.

Scenario 7: You want to protect system and application services against hardware failure.

You can distribute the application and system services to run on nodes configured on different host machines. Even if one of the hosts fails, the application and system services on nodes configured on other hosts continue to provide the service.

Scenario 8: You want to protect system services against disk failure.

When you partition the RulePoint objects to run on application services on nodes configured across multiple host machines, you need to configure additional UM stores to persist the data.

The persistent store provides disk-based storage of messages, where the stores persist the message acknowledgements from the run-time components to the disk. You must have multiple UM stores on separate host machines and each store must have a dedicated disk. UM holds messages in memory until they are written to the disk. If you have multiple UM stores, you can spread the hard disk usage across multiple physical disks. This increases the flexibility for spreading data reception and persistent data load.

Configuring a Topology

Use the **Topology** view to view the run-time components. You can edit the name and properties of a topology created during installation time. You can build a topology for vertical and horizontal scalability, partitioning, capacity, and high availability.

You can configure the following tasks for the topology elements based on your requirements:

- Host. Add, edit, view hosts.
- Node. Add, edit, view, and remove a node.
- Application Services. Add source controllers, event processor, responder controllers, and activity managers.
 - Source controller. Create a source controller in standalone mode, or high-availability mode. Edit, view, and delete a source controller.
 - Event processor. Create an event processor in standalone mode, or high-availability mode. Edit, view, and delete an event processor.
 - Responder controller. Create a responder controller in standalone mode, or high-availability mode. Edit, view, and delete a responder controller.
 - Activity manager. Create or edit an activity manager.
- System Services. Add grid manager, UM store, and UM lbmrd.
 - Grid manager. Add, view, and remove a grid manager.
 - UM store. Add, view, and remove a UM store.
 - UM lbmrd. Add, view, and remove UM lbmrd.

Topology Properties

During installation, you define the properties of a topology, including the database properties, database connection, and SMTP.

After installation, you can edit the topology properties from the **Topology** tab.

The following table describes the properties of a topology:

Properties	Description
jdbcString	URL of the run-time database.
driverClass	Driver class of the run-time database.
username	User name of the run-time database.
password	Password of the run-time database.
maxPoolSize	Maximum pool size of the run-time database connection. The default value is 10.
minPoolSize	Minimum pool size of the run-time database connection. The default is 5.
enable.na.stats	The property that persists the statistics of the host agent to the database. If you set it to true, the property updates the T_AM_HOST_STATUS table of RulePoint_ActivityManager. If you set the value to false, the grid manager does not use this value.
um.store.transport.tcp.low.port	Scans the starting port number of the port range for creating UM topics. The default value is 17500. Note: The port range must be greater than the total number of ports used by responses and sources to ensure successful objects deployment.
um.store.transport.tcp.high.port	Scans the ending port number of the port range for creating UM topics. The default value is 17999. The default range is 499. Note: The port range must be greater than the total number of ports used by responses and sources to ensure successful objects deployment.
np.enabled	Enable emergency alerts.
np.email.1.to	Email address of the recipient.
np.email.1.smtpHost	SMTP host of the mail server.
np.email.1.smtpPort	Port used by the outgoing mail server. Valid values are from 1 to 65535. Default is 25.
np.email.1.from	The email address that the grid manager uses in the From field when sending notification emails. If you leave this field blank, the grid manager uses Administrator@<host name> as the sender.
np.email.1.priority	Sends email for high, critical, medium, and low severities in the system. The priority can be critical or high. Critical information includes scenarios where the design time, runtime, grid manager, or UM store goes down. High priority information includes all the critical scenarios and also includes scenarios where the nodes are switched over, application services and nodes stop functioning, startup and shutdown of the topology, and when the UM store or lbmrd goes down. Do not use medium and low currently.

Properties	Description
np.email.1.verbosity	Send verbose email with detailed statistics. The options are "More" and "Less." If you set it to More, you additionally receive the information along with the stack trace.
np.email.1.username	The user name used for SMTP host authentication upon sending, if required by the outbound mail server.
np.email.1.password	The user password used for SMTP host authentication upon sending, if required by the outbound mail server.
WebService.maxWSDLcache	The number of WSDL parsed objects that a controller can cache. The older WSDL parsed objects are removed from the cache when the number of web service connections exceed the configured max WSDL cache value.

Editing a Topology

You can edit the name and properties of a topology created during installation time.

1. On the **Administrator** tab, click the **Topology** view.
The Topology Model appears in the navigator.
2. Select the topology name, and click the **Edit** button from the menu on the right side.
3. Under **Details**, edit the name and description of the topology.
4. Under **Properties**, edit the topology configurations.
5. Click **Save**.
6. Restart the topology instance to reflect the changes.

Manually Setting Properties in Configuration Files

You can manually change the properties in the configuration files to optimize RulePoint.

The files are located at `RULEPOINT_HOME/conf`.

The following table provides the values that you manually set for the properties in the `am_hibernate.cfg` and `cs.hibernate.cfg` files:

Property	Configuration Setting
hibernate.show_sql	By default, this value is set to false. If you set it to true, it creates a separate log file that lists the queries. Hibernate runs these queries to the database.

The following table provides the values that you manually set for the properties in the `lbmstore.xml` file:

Property	Configuration Setting
repository-size-threshold	The default threshold memory for a topic is 100 MB. You can increase or decrease the memory based on the volume of event flow. If the volume of events is low and the RAM of the machine is less, you can reduce the repository-size-threshold to 50 MB.
repository-size-limit	The default maximum memory is 200 MB. You can proportionally increase or decrease the memory based on the volume of events. If the volume of events is low and the RAM of the machine is less, you can reduce the repository-size-limit value to 100 MB.
repository-disk-file-size-limit	The default disk space utilized is 8 GB for "ORDERED_TOPIC_*." ORDERED_TOPIC_* are topics related to the event processor. The default hard disk space utilization is 2 GB for other topics. You can increase the hard disk space for each topic. The disk space utilized increases with an increase in the number of events that arrive and persist in the disk. If the disk size is less, you can reduce the repository-disk-file-size-limit so that UM store occupies lesser disk space.
source-activity-timeout	If a UM source becomes inactive after the set timeout period, a proxy source is started in the UM store. The default is set to 10 seconds. When the UM source becomes inactive, the UM receiver continues to receive the events as a proxy source comes up, which ensures that no events are lost.
receiver-activity-timeout	If a UM receiver becomes inactive after the set receiver activity timeout period, a proxy receiver is started in the UM store. The default value is set to 168 hours. When the UM receiver stops working, there is no need for the UM receiver to restart.
source-state-lifetime	The default lifetime for a UME source state is set to 2.7 hours. When a source goes out of service, it takes 2.7 hours for the source to stop functioning. You can view the status of the UME source until then in the UME web page.
keepalive interval	You can change the default setting for the interval in which the UM store pings all its registered sources and receiver of 0.5 seconds.
receiver-new-registration-rollback	You can change the default setting for the maximum messages a new receiver can receive from the UM store of 2147483600.

CHAPTER 6

Managing Hosts and Nodes

This chapter includes the following topics:

- [Managing Hosts and Nodes Overview, 53](#)
- [Host, 53](#)
- [Configuring a Host, 54](#)
- [Node, 56](#)
- [Configuring a Node, 56](#)

Managing Hosts and Nodes Overview

If you plan to scale the run-time environment, you can add more hosts and nodes to a default topology. Use the **Topology** view to view the host and node.

First, add more hosts and nodes to a topology, and then distribute the application and system services on the nodes. You have the option to configure the application services on a node to run on standalone mode or high-availability mode. If you configure an application service to run on high-availability mode, you must choose the nodes where you want to run the primary and backup instances. You can configure multiple backup instances and choose to run them on multiple nodes.

You can perform the following tasks to manage hosts and nodes from a topology:

- Host. Add, edit, and view hosts. You can also add nodes, grid manager, UM lbmrd, and UM stores to a host.
- Node. Add, edit, view, and remove a node. You can also add source controllers, event processors, responder controllers, and activity managers to a node.

Host

A host is a physical machine that holds all the RulePoint run-time components. The default port number of the host agent is 19000. A host consists of nodes, application services, and system services.

Configuring a Host

On each host, you can configure one or more nodes, one grid manager, one UM lbmrd, and multiple UM stores. A grid manager manages nodes, application services, and system services in the host.

Adding a Host

When you install RulePoint, the default topology consists of a single host. Use the **Topology** view on the **Administration** tab to add more hosts to a default topology. When you add a host, the host appears in the navigator, and you can view the host properties in the contents panel.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under the **Topology Model** in the navigator, click **Hosts** in the displayed topology default folder.
4. From the **Actions** menu in the right panel, select **Add Host**.
5. Under **Details**, enter the following configuration details:
 - Host Name. The IP address of the machine. Do not use the loopback IP address (127.0.0.1) or the host name.
 - Port. The port number of the host. The port number must be unique. Select a port number between 19000 and 19997.
 - Description. A description of the host.
6. Under **Properties**, enter the following configuration details:
 - na.stats.collector.class. The name of the property class that implements the statistics collector. Enable this property to persist the statistics of the node to the database. The default property is com.informatica.cep.nodestats.NodeStatisticsAggregatorAndPersister.
 - na.stats.collector.default.threads. The number of threads for collecting the statistics for CPU and memory utilization. The default value is 4.
 - na.stats.collector.default.interval. The period for collecting the node statistics. The default is 30 seconds. You can reduce or increase the time based on your requirements.
 - enable.logs.stats. Collects the logs generated by the RulePoint components and stores it in RP_LOG_TRACE table. This information is displayed on the dashboard log viewer. The default is true. You can set it to false if you want to disable the log statistics collection.
7. Click **Save**.

The contents panel displays the configured host in the default topology. The contents panel header also displays the total number of hosts configured.
8. Start the run-time topology.

Editing a Host

You can edit a configured host, change the host details, and then save the changes.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Hosts** in the displayed topology default folder.

The contents panel displays the hosts in the topology.

4. Select the host that you want to edit, and click **Edit** from the menu.
5. Under **Details**, edit the following configuration details:
 - Host Name. The name of the host.
 - Port. The port number of the host. The port number must be unique. Select a port number between 19000 and 19997.
 - Description. A description of the host.
6. Under **Properties**, enter the following configuration details:
 - na.stats.collector.class. The name of the property class that implements the statistics collector. Enable this property to persist the statistics of the node to the database. The default property is com.informatica.cep.nodestats.NodeStatisticsAggregatorAndPersister.
 - na.stats.collector.default.threads. The number of threads for collecting the statistics for CPU and memory utilization. The default value is 4.
 - na.stats.collector.default.interval. The period for collecting the node statistics. The default is 30 seconds. You can reduce or increase the time based on your requirements.
7. Click **Save**.

The contents panel displays the configured node in the default topology. The contents panel header displays the total number of source controllers configured in a topology.
8. Restart the run-time topology.

Viewing Hosts

You can view all the configured hosts in a topology in the contents panel. You can also view the configured details and properties for each host in the topology.

1. On the **Administration** tab, click the **Topology** view.

The navigator displays the **Topology Model** with the default topology configuration.
2. Click **Hosts** in the displayed topology default folder.

The **Lists View** displays the list of hosts in the topology, along with the IP address, the port number, and the last modified details.
3. If you want to view the properties and details of a specific host, select the **Details View**. The right panel displays the details and properties for the selected host.

The following table describes the details and the properties for the selected host:

Property	Description
Details	The details of the selected host, such as: <ul style="list-style-type: none">- Host Name. IP address of the host.- Description. The description of the host.- Port. The port number of the host.- Modified By. The user who last modified the host.- Created Date. The date when the host was last created.- Last Modified Date. The date when the host was last modified.
Properties	The configured properties of the selected host, such as: <ul style="list-style-type: none">- Key. The statistics collector class, threads, and default interval properties for configuring the statistics collector.- Value. The current value of the property.- Initial Value. The initial value of the property.

Deleting a Host

You can delete a configured host from a topology.

Ensure that any system service or node is not connected to the host that you plan to delete. If the topology consists of only one host, you cannot delete that host.

1. On the **Administration** tab, click the **Topology** view.
2. From the **Topology Model** in the navigator, click **Hosts**.
3. In the contents panel, select the host that you want to delete, and select **Delete** from the menu.
A message prompts you to verify if you want to delete the host.
4. Click **OK**.

Node

A node is a java process that can have one or more instances of the run-time components.

The node performs the following functions:

- Manages the controller instances that run on it.
- Interacts with the UM store and the grid manager to provide the communication channel for the run-time components.
- Keeps the application service instances independent and decoupled from each other.
- Manages the lifecycle of components.
- Handles serialization and deserialization of objects.
- Authenticates and authorizes requests to the exposed APIs.
- Provides synchronous communication between the grid manager and controller instances and between the controller instances.
- Provides a uniform API-level support for queue-related operations, such as transparent queuing and dequeuing, irrespective of the underlying transport by the Ultra Messaging layer.
- Responds to the ping requests by the grid manager with heartbeat notifications and run-time metrics of the process, so that the grid manager can make effective load-balancing decisions.

Configuring a Node

During installation, the installer adds a node to a host in a topology. A node can have one or more instances of the run-time components. Each node in a default topology runs the application services, such as the service controllers, the event processor, and the activity manager.

If you configure the run-time components for scalability, you can add one or more nodes to a default topology configuration, and distribute the application services across nodes. When you add more nodes, you can configure them on the same host or across multiple hosts.

If you configure high availability for the run-time components, you can add one or more nodes to a default topology configuration, create primary and backup instances of the application services, and distribute the application services across the nodes.

You can also edit or remove a node from a topology. A host agent runs on each of the hosts in a topology and manages communication between the grid manager and the nodes that run on that particular host.

Use the **Topology** view on the **Administration** tab to manage nodes, configure node properties, and remove nodes from a domain.

Adding a Node

You can add one or more nodes to a default topology. When you add a node, the node appears in the navigator, and you can view the node properties in the contents panel.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Nodes** in the displayed topology default folder.
4. Select **Add Node** from the Actions menu in the right panel.
5. Under **Details**, enter the following configurations:
 - Name. Name of the node.
 - Description. (Optional) Description of the property.
 - Host. The IP address of the host where you want to run the node.
 - Port. The port number of the host. The port number must be unique. Select a port number between 19000 and 19997.
6. Under **Properties**, enter the Java Virtual Machine (JVM) options, which specify the properties of the Java process that supports the node. The default value property is Xmx512m. You can change the value based on your requirement.
7. Click **Save**.

The contents panel displays the configured node in the default topology. The contents panel header displays the total number of nodes configured in a topology.
8. Start the run-time topology.

Example JVM Property

The following JVM option is an example that you can use:

```
-Xmx2048m -verbose:gc -XX:+PrintGCTimeStamps -XX:+PrintGCDetails -Xloggc:jvm_heap_stats_ops_engine.log -XX:+DisableExplicitGC -XX:+HeapDumpOnOutOfMemoryError
```

Editing a Node

You can edit the properties of a node in a topology.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. From the **Topology Model** in the navigator, click **Nodes** in the displayed topology default folder.
4. In the contents panel, select the node that you want to edit, and click **Edit** from the menu on the right.
5. Under **Details**, edit the following configurations:
 - Name. Name of the node.
 - Description. (Optional) Description of the property.

- **Host.** The IP address of the host where you want to run the node.
 - **Port.** The port number of the host. The port number must be unique. Select a port number between 19000 and 19997.
6. Under **Properties**, enter the Java Virtual Machine (JVM) options, which specify the properties of the Java process that support the node. The default value property is Xmx512m.
 7. Click **Save**.
The contents panel displays the configured node in the default topology. The contents panel header displays the total number of nodes configured in a topology.
 8. Start the run-time topology to apply the configured changes.

Viewing Nodes

You can view all the configured nodes in a topology in the contents panel. The contents panel displays the configured details and properties for each node in the topology.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **Nodes** in the displayed topology default folder.
The contents panel displays the list of nodes in the topology in **List View**.
3. If you want to view the details and properties of a specific node, click **Details View**, and select the node that you want to view in the left panel.

The right panel displays the details and properties for the selected node.

The following table describes the details and the properties for the selected node:

Property	Description
Details	<p>The details of the selected node, such as:</p> <ul style="list-style-type: none"> - Name. Name of the node. - Host. IP address of the host. - Port. The port number of the node. - Created By. The name of the user who last created the node. - Modified By. The user who last modified the node. - Created Date. The date when the node was last created. - Last Modified Date. The date when the node was last modified. - Description. The description of the node.
Properties	<p>The configured properties of the selected node, such as:</p> <ul style="list-style-type: none"> - Key. Name of the Java Virtual Machine (JVM) property that runs the java process for the node. - Value. The current value of the property. - Initial Value. The initial value of the property.

Deleting a Node

If the topology consists of only one node, you cannot delete that node.

Note: Ensure that any application service is not running on the node that you plan to delete.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. From the **Topology Model** in the navigator, click **Nodes**.

4. In the contents panel, select the node that you want to delete, and then select **Delete** from the menu displayed on the right.
A message prompts you to verify if you want to delete the node.
5. Click **OK**.
The removed node does not display in the contents panel.
6. Start the run-time topology.

CHAPTER 7

Managing Application Services

This chapter includes the following topics:

- [Application Services Overview, 60](#)
- [Service Controller, 60](#)
- [Configuring a Source Controller, 61](#)
- [Configuring a Responder Controller, 65](#)
- [Event Processor, 69](#)
- [Configuring an Event Processor, 69](#)
- [Activity Manager, 73](#)
- [Configuring the Activity Manager, 73](#)

Application Services Overview

When you install RulePoint, the installation program installs the application services. You can create or manage the application services in the run-time environment.

Application services include the service controllers, event processor, and activity manager, which provide the functionality for the RulePoint run-time operations. You can configure one or more instances of service controllers, event processors, and activity managers. You can choose to add the application services on one or more nodes in the run-time environment based on your requirements.

Service Controller

The service controller is an application service within RulePoint. The service controller handles all message transformations between an inbound data source to the event processor and rule activations from the event processor to the outbound data sources.

Service controllers include the source controllers and responder controllers. Source controllers produce events and responder controllers consume activations.

A service controller is in contact with the system external to the RulePoint application. The service controllers usemarshallers to transform inbound or outbound data, based on a configured set of criteria that a user specifies. Service controllers use data adaptors to either receive data within the data source or send a response.

A service controller performs the following functions:

- Pulls data from a source and transforms it into events.
- Pushes data to a responder and transforms it into a response.
- Manages the lifecycle of controller and responder services.
- Manages all interactions between the various data sources.
- Runs source controller and responder controller services according to a configured service schedule, either dynamic or static.

Configuring a Source Controller

RulePoint installs the default topology, which consists of a single instance of the source controller. You can add one or more source controllers to the default topology configuration.

You can choose to configure the source controllers in standalone mode or high-availability mode. You also have the option to edit or remove a source controller in a topology.

Source Controller Properties

You have the option to enable tracing of source properties in a source controller. When you enable the property, the dashboard displays the lifecycle of the source, the statistics of the events, and error messages.

The following table describes the tracing and other configuration properties in a source controller:

Property	Description
sc.enable.lc.event.tracing	Source lifecycle event tracing. Tracks the lifecycle of source events, such as deploy, undeploy, start, and stop. The dashboard displays the results of the lifecycle of the source events.
sc.enable.source.aggregate.tracing	Source aggregate event tracing. Provides the following statistics for every execution cycle of the source: <ul style="list-style-type: none">- Number of events produced- Execution time of the job The property aggregates the statistics and provides it periodically at the end of the interval, based on the <i>sc.source.aggregate.tracing.interval.period</i> you specify. By default, the interval is set to 60 seconds.
sc.enable.source.service.ha	Source controller high availability. Enables message stability acknowledgements in the UM. The source controller uses these to maintain and persist the state of source's execution. If there is a failover or reassignment, the saved state is used to recover the source to its previous running state. This flag is enabled, by default, for a topology. If you set it to false, the source does not receive the stability acknowledgements and the source controller does not save the running state. For pull-based scheduled sources, if you disable the flags, the state is persisted at the end of the schedule.
sc.enable.source.event.tracing	Source event tracing. Activity manager records every event that the source generates.

Creating a Source Controller in Standalone Mode

You can add one or more source controllers to a default topology. You can choose to configure the source controllers in standalone mode or high-availability mode.

1. Shut down the run-time topology.
2. Add a host to the topology.
3. On the **Administration** tab, click the **Topology** view.
4. Under **Topology Model** in the navigator, click **Source Controller** in the displayed topology default folder.
5. From the **Actions** menu in the upper-right pane, select **Add Source Controller**.
6. Under **Details**, enter the following configuration details for the source controller.
 - a. Under **Name**, provide a name for the source controller.
 - b. Under **Description**, optionally provide a description for the source controller.
 - c. Under **Deployment Mode**, select **Standalone** to run the source controller on a default node.
 - d. Under **Primary Node**, select the node where you want to run the source controller.
7. Under **Properties**, enter the configuration details for the source controller:
 - Type `true` to enable tracing.
 - Type `false` to disable tracing.

For information on the source controller properties, see [“Source Controller Properties” on page 61](#).

8. Click **Save**.

The contents panel displays the configured source controller in the default topology. The contents panel header also displays the total number of source controllers configured, while the lower pane displays the details and properties of the source controller.

9. Start the run-time topology.

Creating a Source Controller in High-Availability Mode

You can add one or more source controllers to a default topology. You can choose to configure the source controllers in high-availability mode if you want multiple backup instances of the source controller.

Note: You must have at least two nodes in a topology to configure high availability.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Source Controller** in the displayed topology default folder.
4. From the **Actions** menu in the upper-right pane, select **Add Source Controller**.
5. Under **Details**, enter the following configuration details for the source controller:
 - a. Under **Name**, provide a name for the source controller.
 - b. Under **Description**, optionally provide a description for the source controller.
 - c. Under **Deployment Mode**, select **High Availability** to create multiple backup instances of the source controller.
 - d. Under **Primary Node**, select the node where you want to run the primary source controller instance.
 - e. Under **Backup Node**, select the node where you want to run the backup instance of the source controller.

You can choose multiple nodes to create multiple backup instances of the source controllers.

6. Under **Properties**, enter the configuration details for the source controller:

- Type `true` to enable tracing.
- Type `false` to disable tracing.

For information on the source controller properties, see [“Source Controller Properties” on page 61](#).

7. Click **Save**.

The contents panel displays the newly configured source controller in the default topology. The contents panel header also displays the total number of source controllers configured, while the lower pane displays the details and properties of the source controller.

8. Start the run-time topology.

Editing a Source Controller

You can edit a configured source controller. You have the option to change the source controller to standalone mode or high-availability mode, or reassign the source controllers to different nodes based on your requirements.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Source Controller** in the displayed topology default folder.
4. Select the source controller in the contents panel that you want to edit, and click **Edit** from the menu.
5. Under **Details**, edit the following properties:
 - a. Change the name of the source controller, and the description of the source controller.
 - b. Under **Deployment Mode**, change the deployment mode for the source controller to Standalone or High availability:
 - c. If you select **Standalone**, select the node where you want to run the source controller.
 - d. If you select **High Availability**, choose the node where you want to run the primary and secondary instance:
 - Under **Primary Node**, select the node where you want to run the primary instance of the source controller.
 - Under **Backup Node**, select the node where you want to run the backup instance of the source controller.
6. Under **Properties**, enter the configuration details for the source controller:

For information on the source controller properties, see [“Source Controller Properties” on page 61](#).
7. Click **Save**.

The contents panel displays the edited source controller in the default topology.
8. Start the run-time topology.

Viewing a Specific Source Controller

You can view a specific source controller to view its details and properties.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **Source Controller** in the displayed topology default folder.

3. In the contents panel, select the source controller that you want to view.

The Details View displays information for the selected source controller.

The following table describes the details and the properties for each source controller:

Property	Description
Details	The details of the selected source controller, such as: <ul style="list-style-type: none">- Name. Name of the source controller.- Description. The description of the source controller.- Created By. The user who last created the source controller.- Modified By. The user who last modified the source controller.- Created Date. The date when the source controller was last created.- Last Modified Date. The date when the source controller was last modified.
Properties	The properties of the selected source controller, such as: <ul style="list-style-type: none">- Key. Name of the Java Virtual Machine (JVM) property that runs the java process for the source controller.- Value. The current value of the java property.- Initial Value. The initial value of the java property.

Viewing All Source Controllers

You can view all the configured source controllers in the contents panel. You can also sort the source controllers by name or description.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **Source Controller** in the displayed topology default folder.

The contents panel displays the list of source controllers in the topology. By default, the contents panel displays the source controllers sorted alphabetically by name.

Deleting a Source Controller

You can remove a source controller from a topology configuration.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Source Controller** in the displayed topology default folder.
4. In the contents panel, select the source controller that you want to delete, and click **Delete** from the menu displayed on the right.

A message prompts you to verify if you want to delete the source controller.

5. Click **OK**.

The removed source controller does not display in the contents panel.

6. Start the run-time topology.

Configuring a Responder Controller

RulePoint installs the default topology, which consists of a single instance of the responder controller. You can add one or more responder controllers to the default topology configuration.

You can choose to configure the responder controllers in standalone mode or high-availability mode. You can also edit or remove a responder controller in a topology.

Responder Controller Properties

You have the option to enable tracing of responder properties in a responder controller. When you enable the property, the dashboard displays the lifecycle of the responder, the statistics of the events, and error messages.

The following table describes the tracing and other configuration properties that you can enable in a responder controller:

Property	Description
<code>sc.enable.response.event.tracing</code>	Response event tracing. Activity manager records the activity for every response received.
<code>sc.enable.lc.event.tracing</code>	Responder lifecycle event tracing. Activity manager tracks the lifecycle of responder events, such as deploy, undeploy, start, and stop. If tracing is enabled, the dashboard displays the results of the lifecycle of the responder events.
<code>sc.enable.response.aggregate.tracing</code>	<p>Response aggregate event tracing. Provides the following statistics for every execution cycle of the response:</p> <ul style="list-style-type: none">- Number of events produced- Execution time of the job <p>The property aggregates the statistics and provides it periodically at the end of the interval, based on the <code>sc.response.aggregate.tracing.interval.period</code> you specify. By default, the interval is set to 60 seconds.</p>
<code>sc.enable.responder.service.ha</code>	<p>Responder controller high availability. Enables explicit acknowledgements on delivering activations from the UM to the responder controller.</p> <p>After the responder controller dispatches the activations, it sends an acknowledgement to the messaging layer of the successful delivery. If the responder shuts down without delivering the activations, all the unacknowledged activations is re-delivered after failover.</p>
<code>sc.enable.response.event.tracing</code>	Response event tracing. Activity manager records the activity for every response received.
<code>sc.response.aggregate.tracing.interval.period</code>	Response aggregate tracing interval. The time interval that the responder controller periodically sends the aggregate response statistics to the activity manager. The default interval is 60 seconds.
<code>HttpServices.encodeQueryString</code>	Enables URI encoding of the target URL. By default, this is disabled.

You can override the default values in the `sc.properties` file in the `<RULEPOINT_HOME>/conf` directory. While doing this, you need to add the properties after the last line in the `sc.properties` file.

The following table describes the properties that you can set for a response:

Property	Description
sc.response.queue.size	The pool length for the number of activations that will be queued. The default size is 1000.
sc.response.initial.pool.size	The initial size of the response execution thread pool. The default size is 10.
sc.response.max.pool.size	The maximum size of the response execution thread pool. The default size is 20.
sc.response.pool.keep.alive.time	The period that the response execution thread is kept idle. The default is 300 seconds.

Creating a Responder Controller in Standalone Mode

You can add one or more responder controllers to a default topology and choose to configure the responder controllers in standalone mode. In a standalone mode, you cannot have a backup instance of the responder controller.

Note: You must have at least two nodes in a topology.

1. Shut down the run-time topology before you start the configuration.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Responder Controller** in the displayed topology default folder.
4. From the **Actions** menu in the upper-right pane, select **Add Responder Controller**.
5. Under **Details**, enter the following configuration details for the responder controller.
 - a. Under **Name**, provide a name for the responder controller.
 - b. Under **Description**, optionally provide a description for the responder controller.
 - c. Under **Deployment Mode**, select **Standalone** to run the responder controller on a default node.
 - d. Under **Primary Node**, select the node where you want to run the responder controller.
6. Under **Properties**, enter the configuration information for the responder controller:
 - To enable tracing, type `true`.
 - To disable tracing, type `false`.

For information on the configuration properties, see [“Responder Controller Properties” on page 65](#).

7. Click **Save**.

The contents panel displays the configured responder controller in the default topology. The contents panel header also displays the total number of responder controllers configured, while the lower panel displays the details and properties of the responder controller.

8. Start the run-time topology.

Creating a Responder Controller in High-Availability Mode

You can add one or more responder controllers to a default topology, and choose to configure the responder controllers in high-availability mode. In this mode, you can configure multiple backup instances for each responder controller. You can choose the nodes where you want to run the primary and secondary instances.

Note: You must have at least two nodes in a topology.

1. Shut down the run-time topology before you start the configuration.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Responder Controller** in the displayed topology default folder.
4. From the **Actions** menu in the upper-right pane, select **Add Responder Controller**.
5. Under **Details**, enter the following configuration details for the responder controller:
 - a. Under **Name**, provide a name for the responder controller.
 - b. Under **Description**, optionally provide a description for the responder controller.
 - c. Under **Deployment Mode**, select **High Availability** to create backup instances of the responder controller.
 - d. Under **Primary Node**, select the node where you want to run the primary responder controller instance.
 - e. Under **Backup Node**, select the node where you want to run the backup instance of the responder controller.

You can choose multiple nodes to create multiple backup instances of the responder controllers.

6. Under **Properties**, enter the configuration details for the responder controller:
 - To enable tracing, type `true`.
 - To disable tracing, type `false`.

For information on the configuration properties, see ["Responder Controller Properties" on page 65](#).

7. Click **Save**.

The contents panel displays the configured responder controller in the default topology. The contents panel header also displays the total number of responder controllers configured, while the details view displays the details and properties of the responder controller.

8. Start the run-time topology.

Editing a Responder Controller

You can edit a configured responder controller. You have the option to change the responder controller to standalone or high-availability mode, or reassign the responder controllers to different nodes based on your requirements.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Responder Controller** in the displayed topology default folder.
4. Select the event processor in the contents pane, and then select **Edit** from the menu.

5. Under **Details**, edit the following properties:
 - Change the name and the description of the responder controller.
 - Under **Deployment Mode**, change the deployment mode for the responder controller to standalone or high availability.
 - Select the nodes where you want to run the responder controller.
6. Under **Properties**, enter the configuration details. Choose to enable or disable the tracing for source events.

For information on the configuration properties, see [“Responder Controller Properties” on page 65](#).
7. Click **Save** to save the changes.

The contents panel displays the edited responder controller in the default topology.
8. Start the run-time topology.

Viewing a Specific Responder Controller

You can view a specific responder controller to view its details and properties.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **Responder Controller** in the displayed topology default folder.
3. In the contents panel, select the responder controller that you want to view.

The details view displays information for the selected responder controller.

The following table describes the details and the properties for each responder controller:

Property	Description
Details	The details of the selected responder controller, such as: <ul style="list-style-type: none">- Name. Name of the responder controller.- Description. The description of the responder controller.- Created By. The user who last created the responder controller.- Modified By. The user who last modified the responder controller.- Created Date. The date when the responder controller was last created.- Last Modified Date. The date when the responder controller was last modified.
Properties	The properties of the selected responder controller, such as: <ul style="list-style-type: none">- Key. Name of the Java Virtual Machine (JVM) property that runs the java process for the responder controller.- Value. The current value of the java property.- Initial Value. The initial value of the java property.

Viewing All Responder Controllers

You can view all the configured responder controllers in the contents panel. You can also sort the responder controllers by name or description.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **Responder Controller** in the displayed topology default folder.

The contents panel displays the list of configured responder controllers in the topology, sorted alphabetically by name.

Deleting a Responder Controller

You can remove a responder controller from a topology configuration.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Responder Controller** in the displayed topology default folder.
4. In the contents panel, select the responder controller that you want to delete, and then select **Delete** from the menu.

A message prompts you to verify if you want to delete the responder controller.

5. Click **OK**.

The removed responder controller does not display in the contents panel.

6. Start the run-time topology.

Event Processor

The event processor is the core component of RulePoint, and processes the configured rules in the run-time environment.

The event processor manages the lifecycle of all objects deployed on it. It correlates an incoming event from a source controller with past events by using predefined rules for that event topic. Based on the rule condition, the event processor can store events in the local cache of the rule, so that it can later correlate the events with future events. Events in the local cache are removed after expiry or after the events activate the rule. The event processor generates responses whenever the rule matches the event.

Configuring an Event Processor

RulePoint installs the default topology, which consists of a single instance of the event processor. You can add one or more event processors to the default topology configuration.

You can choose to configure the event processors in standalone mode or high-availability mode. In a standalone mode, the event processor will not have a backup event processor process. In a high-availability mode, you can configure multiple backup instances for an event processor. You can create multiple instances of event processors for both standalone and high-availability mode, so that you can split the load traffic between the instances.

You also have the option to edit or remove an event processor in a topology.

Event Processor Properties

You can view the properties for the event processor in **Details View**. Select the event processor in the **Contents** panel to view the properties.

The following table describes the event processor configuration properties:

Property	Description
eg.enable.instrumentation.global	Controls whether an event processor can send instrumentation data to the activity manager during the rule writing process. The default value is true.
eg.enable.ha	Enables the primary engine to send the source event stream to the secondary engines and periodically sends the primary engine's state to the secondary engines. If there is a failover, the secondary engine starts processing, without losing the primary engine's state. The default is true.
eg.event.storage.capacity	Maximum number of events that an event processor can store before blocking new event injection. The default value is 500000.
eg.maximum.event.age	Maximum amount of time, in seconds, that an event remains alive in the event processor for evaluation by a particular rule. The default value is 1800.
eg.simple.executor.threadpool.size	Number of threads that an event processor can run for simple conditions of rules. The default value is 8.
eg.complex.executor.thread.pool.size	Number of threads to run complex rule conditions. The default value is 50.

Creating an Event Processor in Standalone Mode

You can add one or more event processors to a default topology, and choose to configure the event processors in standalone mode.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Event Processor** in the displayed topology folder.
4. From the **Actions** menu in the right pane, select **Add Event Processor**.
5. Under **Details**, enter the following configuration details for the event processor.
 - a. Under **Name**, provide a name for the event processor.
 - b. Under **Description**, optionally provide a description for the event processor.
 - c. Under **Deployment Mode**, select **Standalone** to run the event processor on a default node.
 - d. Under **Primary Node**, select the node where you want to run the event processor.
6. Under **Properties**, enter the values for the event processor and related configuration details.
7. Click **Save**.

The contents panel displays the configured event processor in the default topology. The contents panel header also displays the total number of event processors configured. The details view displays the details and properties of the event processor.

8. Start the run-time topology.

Creating an Event Processor in High-Availability Mode

You can add one or more event processors to a default topology, and choose to configure the event processors in high-availability mode. Each event processor can have multiple backup instances.

Note: You must have at least two nodes in a topology to configure an event processor in high-availability mode.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Event Processor** in the displayed topology folder.
4. From the **Actions** menu in the right pane, select **Add Event Processor**.
5. Under **Details**, enter the following configuration details for the event processor:
 - a. Under **Name**, provide a name for the event processor.
 - b. Under **Description**, optionally provide a description for the event processor.
 - c. Under **Deployment Mode**, select **High Availability** to create backup instances of the event processor.
 - d. Under **Primary Node**, select the node where you want to run the primary event processor instance.
 - e. Under **Backup Node**, select the node where you want to run the backup instance of the event processor.

You can choose multiple nodes to create multiple backup instances of the event processors.

6. Under **Properties**, enter the values for the event processor and related configuration details.
7. Click **Save**.

The contents panel displays the configured event processor in the default topology. The contents panel header also displays the total number of event processors configured. The details view displays the details and properties of the event processor.

8. Start the run-time topology.

Editing an Event Processor

You can edit a configured event processor. You have the option to change the event processor to standalone or high availability mode, or reassign the event processors to different nodes based on your requirements.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Event Processor** in the displayed topology default folder.
4. Select the event processor in the contents pane, and then select **Edit** from the menu.
5. Under **Details**, edit the following properties:
 - a. Change the name of the event processor, and the description of the event processor.
 - b. Under **Deployment Mode**, change the deployment mode for the event processor to standalone or high availability:
 - c. If you have selected **Standalone**, select the node where you want to run the event processor.
 - d. If you have selected **High Availability**, choose the nodes where you want to run the primary and secondary instances.
6. Under **Properties**, edit the values for the event processor and related configuration details.
7. Click **Save** to save the configured changes.

The contents panel displays the edited event processor in the default topology.

8. Start the run-time topology.

Viewing a Specific Event Processor

You can view a specific event processor to view its details and properties.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **Event Processor** in the displayed topology default folder.
3. In the contents panel, select the event processor that you want to view.

The details view displays the details and properties for the selected event processor.

The following table describes the details and the properties for each event processor:

Property	Description
Details	The details of the selected event processor, such as: <ul style="list-style-type: none">- Name. Name of the event processor.- Description. The description of the event processor.- Created By. The user who last created the event processor.- Modified By. The user who last modified the event processor.- Created Date. The date when the event processor was last created.- Last Modified Date. The date when the event processor was last modified.
Properties	The properties of the selected event processor, such as: <ul style="list-style-type: none">- Key. Name of the Java Virtual Machine (JVM) property that runs the java process for the event processor.- Value. The current value of the java property.- Initial Value. The initial value of the java property.

Viewing All Event Processors

You can view all the configured event processors in the contents panel. You can also sort the event processors by name or description.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **Event Processor** in the displayed topology default folder.

The contents panel displays the list of configured event processors sorted alphabetically by name.

Deleting an Event Processor

You can remove an event processor from a topology configuration.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.

The navigator displays the Topology Model with the default topology configuration.
3. Click **Event Processor** in the displayed topology default folder.
4. In the contents panel, select the event processor that you want to delete, and then select Delete from the menu.

A message prompts you to verify if you want to delete the event processor.

5. Click **OK**.
The deleted event processor does not display in the contents panel.
6. Start the run-time topology.

Activity Manager

The activity manager records and stores management information and activity of all the run-time components. It records the lifecycle of all objects.

The activity manager in RulePoint performs the following functions:

- Tracks the system information, such as the CPU and memory details.
- Stores the data store configuration details, topic, source, and responder details.
- Stores and retrieves the configured deployment data.
- Persists the lifecycle of events and execution details from the event processor.
- Stores events generated by all sources, and uses these events for reporting and analytics purpose.
- Stores the event lifecycle data, so that it can replay events that the event processor has not completely processed.
- Displays all information that it tracks and stores on the dashboard.

Note: The data stored in the activity manager, such as the events and alerts are auto purged every two hours if the count of alerts and events in the table is more than 1,00,000. The data for the sourceExecutionStats, ResponseExecutionStats, RuleExecutionStats, AnalyticExecutionStats, TopicStats, HostStats, and ArtifactHistory is purged from the database every 24 hours.

Configuring the Activity Manager

RulePoint installs the default topology, which consists of a single instance of the activity manager.

You can add one or more activity managers to the default topology configuration to distribute the load. You can also edit or delete an activity manager.

Activity Manager Properties

You can view the properties and details for the activity manager when you select **Details View**.

Select the activity manager in the contents panel to view the properties. When you add an activity manager to a topology, you can choose to type `true` to enable the property, or type `false` to disable the property. You also have the option to edit an activity manager in a topology.

The following table describes the configuration properties for the activity manager:

Property	Description
am.receive.events	Activity manager receives events generated from a source. The default is true.
am.events.persist.enablebatch	Activity manager persists the events in a batch. The default is true.
am.receive.alerts	Activity manager receives responder alerts. The default is true.
am.alerts.persist.enablebatch	Activity manager persists the alerts in a batch. The default is true.
am.receive.ruleTrace	Activity manager traces rules. The default is true.
am.ruleTrace.persist.enablebatch	Activity manager persists the rule tracing in a batch. The default is false.
am.receive.artifactDetails	Activity manager receives the source execution aggregates. The default is true.
am.artifactDetails.persist.enablebatch	Activity manager persists the source execution aggregates in a batch. The default is false.
am.receive.sourceAggregates	Activity manager receives the responder execution aggregates. The default is true.
am.sourceAggregates.persist.enablebatch	Activity manager persists the source execution aggregates in a batch. The default is false.
am.receive.responseAggregates	Activity manager receives the responder execution aggregates. The default is true.
am.responseAggregates.persist.enablebatch	Activity manager persists the responder execution aggregates in a batch. The default is false.
am.receive.ruleAggregates	Activity manager receives the rule execution aggregates. The default is true.
am.ruleAggregates.persist.enablebatch	Activity manager persists the rule execution aggregates in a batch. The default is false.
am.db.min.pool.size	The minimum database pool size that the activity manager must use. The default is 5.
am.db.max.pool.size	The maximum database pool size that the activity manager must use. The default is 15.
am.db.batch.max.queue.size	The maximum number of objects that the activity manager keeps in the cache before it persists it into the database. The default number of objects is 1000.
am.db.batch.persist.interval	The time interval in seconds that the activity manager checks for available objects to persist to the database when you enable a batch. The default is 30.

Property	Description
am.db.batch.send.size	The number of objects to send together to the database when you enable a batch event. The default is 100.
am.db.batch.retry.timeout	The amount of time, in seconds, that the activity manager waits before retrying when the batch queue is full. The default is 2.

Adding an Activity Manager

The installer installs the default topology, which consists of a single activity manager. You can add more activity managers to a default topology.

1. Shut down the run-time topology.
2. Add a host to the topology before you add an activity manager.
3. On the **Administration** tab, click the **Topology** view.
4. Under **Topology Model** in the navigator, click **Activity Manager** in the displayed topology default folder.
5. From the **Actions** menu in the upper-right panel, select **Add Activity Manager**.
6. Under **Details**, provide the following information:
 - a. Under **Name**, type a name for the activity manager.
The name must not start with numbers or spaces.
 - b. Under **Description**, optionally, provide a description for the activity manager.
 - c. Under **Node**, select the node where you want to add the activity manager.
7. Under **Properties**, provide the configuration details.
For more information on the configuration properties for the activity manager, see [“Activity Manager Properties” on page 73](#).
For more information to set the properties for an activity manager instance to either receive only events, activations, or alerts, see [“Creating Multiple Instances of Activity Manager” on page 76](#).
8. Click **Save**.
9. Start the run-time topology.

Editing an Activity Manager

You can edit the name and properties of an activity manager. You can also choose to run the activity manager on a different node.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Activity Manager** in the displayed topology default folder.
4. In the contents panel, select the activity manager that you want to edit, and then select **Edit** from the menu.
5. Under **Details**, edit the following information:
 - a. Under **Name**, type a name for the activity manager.
The name must not start with numbers or spaces.

- b. Under **Description**, optionally, provide a description for the activity manager.
 - c. Under **Node**, select the node where you want to add the activity manager.
6. Under **Properties**, edit the configuration details.
For more information on the configuration properties for the activity manager, see [“Activity Manager Properties” on page 73](#).
7. Click **Save**.
8. Start the run-time topology.

Deleting an Activity Manager

You cannot delete if you have only one activity manager in a topology.

1. Shut down the run-time topology.
2. On the **Administrator** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, select **Activity Manager**.
4. In the contents panel, select the activity manager that you want to delete, and select **Delete** from the menu.
A message appears that prompts you if you want to delete.
5. Click **OK**.
6. Start the run-time topology.

Creating Multiple Instances of Activity Manager

The activity manager receives events from the source controller, engine processor, and responder controller, and loads the information to the corresponding tables in the database. The information from the database is displayed on the dashboard.

In scenarios where large number of events flow into the system, the activity manager may not be able to handle the load. To reduce the load on the activity manager, you can create multiple activity manager instances that cater to the need of only one set of controllers. For example, you can create an activity manager instance for all source controllers, event processors, or responder controllers, irrespective of whether the controller is configured in standalone or high availability mode.

Setting Properties for Activity Manager Instances

When you create an additional instance of activity manager, you need to set the properties for the activity manager if you want the activity manager to only receive data from a specific type of controller.

Set the following properties in the activity manager to only receive events from the source controller:

Property	Property Setting
am.receive.events	true
am.receive.sourceAggregates	true
am.receive.responseAggregates	false
am.responseAggregates.persist.enablebatch	false

Property	Property Setting
am.receive.ruleAggregates	false
am.ruleAggregates.persist.enablebatch	false

Set the following properties in the activity manager to only receive activations from the event processor:

Property	Property Setting
am.receive.ruleAggregates	true
am.ruleAggregates.persist.enablebatch	true
am.receive.responseAggregates	false
am.responseAggregates.persist.enablebatch	false
am.receive.events	false
am.receive.sourceAggregates	false

Set the following properties in the activity manager to only receive alerts from the responder controller:

Property	Property Setting
am.receive.events	true
am.receive.sourceAggregates	true
am.receive.responseAggregates	false
am.responseAggregates.persist.enablebatch	false
am.receive.ruleAggregates	false
am.ruleAggregates.persist.enablebatch	false

CHAPTER 8

Managing System Services

This chapter includes the following topics:

- [System Services Overview, 78](#)
- [Grid Manager, 78](#)
- [Configuring a Grid Manager, 79](#)
- [UM lbmrd, 83](#)
- [Configuring the UM lbmrd, 83](#)
- [UM Store, 85](#)
- [Configuring a UM Store, 85](#)

System Services Overview

When you install RulePoint, the installation program installs the system services that manage the run-time environment.

System services include the grid manager, UM store, and UM lbmrd. You can create or manage the system services in the run-time environment. You can configure one or more instances of the system services, and run them on one or more nodes in the run-time environment.

Grid Manager

The grid manager controls how the various RulePoint entities work in the run-time environment.

At a broad level, the grid manager performs the following functions:

- Manages all configurations of the run-time components.
- Manages interactions between the various run-time components.
- Starts the passive grid manager processes.
- Starts the node processes.
- Starts the umstored and lbmrd processes.
- Authenticates and authorizes requests to the exposed APIs.
- Checks the license file.

- Deploys or undeploys objects according to requests received from the design UI and maintains the run-time system.
- Provides high availability and failover of objects by replicating them across multiple instances. The grid manager controls switching from the primary to the secondary controller instance if the primary fails.
- Monitors the health of the nodes by pinging nodes at regular intervals to verify if the nodes are working.
- Shares the state of the run-time system with the passive counterpart to ensure a seamless failover when the primary grid manager in high-availability mode is down.
- Checks the health of host agents.
- Starts the backup grid manager processes as designated on nodes.
- Starts nodes, starts the backup grid manager processes on nodes, and adds controllers on nodes, as designated.
- Brings up the umestored and lbmrd if they go down, and monitors their health.
- Sends emails when any component fails, if you have configured this option during installation.

Configuring a Grid Manager

RulePoint installs the default topology, which consists of a single instance of the grid manager. You can add one or more grid managers to a topology.

If you plan to configure multiple instances of the grid manager for high availability, you must distribute them in different hosts. When you start RulePoint in a particular host, the grid manager in that host becomes the primary instance. When the primary instance fails, one of the multiple grid manager instances takes over the role of the primary.

You also have the option to edit or remove a grid manager in a topology.

Grid Manager Configuration Properties

When you create a grid manager, you must enter the values for each of the properties through the user interface. You can also edit the configuration properties of a grid manager.

The following table describes the properties for configuring a grid manager:

Property	Description
rpserver.jvm.options	The JVM options for the java process that hosts the grid manager. The default is -Xmx512m.
rpserver.topology.operation.thread.pool.size	The maximum number of threads to use for various functions that bring up or bring down a topology. The default is 10.
rpserver.topology.operation.task.queue.size	The maximum number of topology element tasks that the queue can hold while bringing up or bringing down a topology. The default is 10.
rpserver.deploy.operations.thread.pool.size	The number of threads to use for deployment-related operations of objects. The default is 5.
rpserver.probe.threadpool.size	The number of threads to use to verify if the hosts, the nodes, and the grid manager are working. The default is 5.

Property	Description
rpserver.eu.pingInterval	The ping interval, in seconds, to verify if the nodes are working. The default is 10 seconds.
rpserver.eu.retry.count	The number of retry attempts to ping the node before assuming that the node is down. The default is 2.
rpserver.server.pingInterval	The grid manager ping interval, in seconds. The default is 10 seconds.
rpserver.server.retry.count	The grid manager ping retry count before assuming that the grid manager is down. The default is 2.
pserver.node.pingInterval	The interval, in seconds, for pinging the host agent. The default is 15 seconds.
rpserver.node.retry.count	The retry count for the host agent before assuming that the host agent has failed. The default is 2.
rpserver.db.operation.failure.retry.delay	The interval, in seconds, to retry a failed database operation. Setting the retry interval takes care of temporarily database failures. The default is 5 seconds.
rpserver.db.operation.failure.retry.count	The number of times to retry a failed database operation before assuming that the operation has failed. The default is 2.
rpserver.db.pingInterval	The interval, in seconds, to check the database availability after it goes down. The default is 10 seconds.
rpserver.eu.start.timeout	Timeout, in seconds, to start a node. If the node does not start within the specified time, the grid manager assumes that the node has failed. The default timeout is 60 seconds.
rpserver.eu.stop.timeout	Timeout, in seconds, to stop a node. If the node does not stop within the specified time, the grid manager stops the node process. The default timeout is 30 seconds.
rpserver.controller.start.timeout	Timeout, in seconds, to start controllers in a node. If the controller does not start within the specified time, the grid manager assumes that the controller has failed. The default start timeout is 60 seconds.
rpserver.controller.stop.timeout	Timeout, in seconds, to stop controllers in the node. If the controller does not stop within the specified time, the grid manager removes the controller from the associated node. The default timeout is 30 seconds.
rpserver.umstore.start.timeout	Timeout, in seconds, to start a UM store. If the store does not start within the specified time, the grid manager assumes that the UM store has failed. The default start timeout is 300 seconds.
rpserver.umlbrd.start.timeout	Timeout, in seconds, to start the UM lbrd. If the lbrd does not start within the specified time, the grid manager assumes that the UM lbrd has failed. The default start timeout is 10 seconds.

Adding a Grid Manager to a Default Topology

The default topology consists of a single instance of the grid manager. You can add one or more grid managers to the default topology configuration, however, you can add only one grid manager to a host.

1. Shut down the run-time topology.
2. Add an additional host to a default topology where you want to add the grid manager.
3. On the **Administration** tab, click the **Topology** view.
4. Under **Topology Model** in the navigator, click **Grid Manager** in the displayed topology default folder.
5. From the **Actions** menu in the upper-right pane, select **Add Grid Manager**.
6. Under **Details**, enter the following information:
 - a. Under **Name**, provide a name for the grid manager.
The name must not start with numbers or whitespace.
 - b. Under **Description**, optionally provide a description for the grid manager.
 - c. Under **Host**, select the IP address of the host from the list, where you want to run the grid manager.
 - d. Under **Port**, enter a port number for the grid manager. The port number must be unique. Select a port number between 19000 and 19997.
7. Under **Properties**, enter the configuration properties.
For more information, see ["Grid Manager Configuration Properties" on page 79](#)
8. Click **Save**.
The contents panel displays the configured grid manager in the default topology. The contents panel header also displays the total number of grid managers configured. The details view displays the details and properties of the grid manager.
9. Start the run-time topology.

Editing a Grid Manager

You can edit the properties of a configured grid manager. You can also choose to deploy the configured grid manager on another host based on your requirements.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Grid Manager** in the displayed topology default folder.
4. In the contents pane, select the grid manager that you want to edit, and click **Edit** from the menu.
5. Under **Details**, edit the following information:
 - a. Under **Name**, provide a name for the grid manager.
 - b. Under **Description**, optionally provide a description for the grid manager.
 - c. Under **Host**, enter the IP address of the host where you want to deploy the grid manager.
 - d. Under **Port**, enter the port number for the grid manager.
The port number must be unique. Select a port number between 19000 and 19997.
6. Under **Properties**, edit the properties for the grid manager.
7. Click **Save**.
The contents panel displays the newly edited grid manager in the default topology. The details view displays the newly edited details and properties of the grid manager.

8. Start the run-time topology.

Viewing a Specific Grid Manager

You can view a specific grid manager to view its details and properties.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **Grid Manager** in the displayed topology default folder.
3. In the contents panel, select the grid manager that you want to view, and click **Details View**.

You can view the details and properties for the selected grid manager.

The following table describes the details and the properties for each grid manager:

Property	Description
Details	The details of the selected grid manager, such as: <ul style="list-style-type: none">- Name. Name of the grid manager.- Description. The description of the grid manager.- Created By. The user who last created the grid manager.- Modified By. The user who last modified the grid manager.- Created Date. The date when the grid manager was last created.- Last Modified Date. The date when the grid manager was last modified.
Properties	The properties of the selected grid manager, such as: <ul style="list-style-type: none">- Key. Name of the Java Virtual Machine (JVM) property that runs the java process for the grid manager.- Value. The current value of the java property.- Initial Value. The initial value of the java property.

Viewing All Grid Managers

You can view all the configured grid managers in the contents panel. You can also sort the grid managers by name or description.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **Grid Manager** in the displayed topology default folder.

The contents panel displays the list of grid managers in the topology. By default, the contents panel displays the grid managers sorted alphabetically by name.

Deleting a Grid Manager

You can remove a grid manager from a topology configuration. Your topology must have more than one grid manager to carry out the delete operation.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **Grid Manager** in the displayed topology default folder.
4. In the contents panel, select the grid manager that you want to delete, and click **Delete** from the menu.

A message prompts you to verify if you want to delete the grid manager.

5. Click **OK**.
6. Start the run-time topology.

UM lbmrd

Ultra Messaging Latency Busters Messaging Resolver Daemon (UM lbmrd) provides the address resolver services for messaging in RulePoint.

The run-time components are configured to use unicast topic resolution. The store configuration file consists of the unicast topic resolution daemon (lbmrd) options.

Configuring the UM lbmrd

RulePoint installs the default topology, which consists of a single instance of the lbmrd. You can add one or more lbmrds to a default topology configuration.

Running multiple instances of lbmrd allows your applications to continue operation in the face of an lbmrd failure. Your applications services continue to send topic resolution messages as usual.

You also have the option to edit or remove an lbmrd in a topology.

Adding a UM lbmrd to a Default Topology

You can add one or more UM lbmrds to a default topology. A default topology consists of a single UM lbmrd. Each host can have only one UM lbmrd.

1. Shut down the run-time topology.
2. Add a host to the topology where you want to add the UM lbmrd.
3. On the **Administration** tab, click the **Topology** view.
4. Under **Topology Model** in the navigator, click **UM lbmrd** in the displayed topology default folder.
5. From the **Actions** menu in the upper-right pane, select **Add UM lbmrd**.
6. Under **Details**, enter the following configurations:
 - a. Under **Name**, provide a name for the UM lbmrd.
The name must not start with numbers or space.
 - b. Under **Description**, optionally provide a description for the UM lbmrd.
 - c. Under **Host**, enter the IP address of the host where you want to run the UM lbmrd.
 - d. Under **Port**, enter the port number for the UM lbmrd. The port number must be unique. Select a port number between 22000 and 22999.
 - e. Under **Commandline Options**, enter the additional parameters for UM lbmrd.
7. Click **Save**.

The contents panel displays the configured UM lbmrd in the default topology. The contents panel header also displays the total number of UM lbmrds configured. The details view displays the details and properties of the UM lbmrd.
8. Start the run-time topology.

Editing a UM Ibmrdr

You can edit the properties of a configured UM Ibmrdr. You can also choose to run the configured UM Ibmrdr on another host.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **UM Ibmrdr**.
4. Select the UM Ibmrdr that you want to delete, and select **Edit** from the menu.
5. Edit the configurations for the UM Ibmrdr.
6. Click **Save**.
7. Start the run-time topology.

Viewing a Specific UM Ibmrdr

You can view a specific UM Ibmrdr for its details and properties.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **UM Ibmrdr** in the displayed topology default folder.
3. In the contents panel, select the UM Ibmrdr that you want to view, and click **Details View**.

You can view the details and properties for the selected UM Ibmrdr.

The following table describes the details and the properties for each UM Ibmrdr:

Property	Description
Details	The details of the selected UM Ibmrdr, such as: <ul style="list-style-type: none">- Name. Name of the UM Ibmrdr.- Description. The description of the UM Ibmrdr.- Host. The host IP address.- Port. The port number.- Commandline Options. Any additional configurations for UM Ibmrdr.- Created By. The user who last created the UM Ibmrdr.- Modified By. The user who last modified the UM Ibmrdr.- Created Date. The date when the UM Ibmrdr was last created.- Last Modified Date. The date when the UM Ibmrdr was last modified.
Properties	The properties of the selected UM Ibmrdr, such as: <ul style="list-style-type: none">- Key. Name of the Java Virtual Machine (JVM) property that runs the java process for the UM Ibmrdr.- Value. The current value of the java property.- Initial Value. The initial value of the java property.

Viewing All UM Ibmrdr

You can view all the configured UM Ibmrdrs in the contents panel. You can also sort the UM Ibmrdrs by name or description.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **UM Ibmrdr** in the displayed topology default folder.

The contents panel displays the list of UM Ibmrdrs in the topology, sorted alphabetically by name.

Deleting a UM lbmrd

You must have more than one UM lbmrds in a topology to carry out the delete operation.

1. Shut down the run-time topology.
2. On the **Administrator** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, select UM lbmrd.
4. In the contents panel, select the UM lbmrd that you want to delete, and click **Delete** from the menu.
A message appears that prompts you if you want to delete.
5. Click **OK**.
6. Start the run-time topology.

UM Store

UM store is the persistent store daemon, which is based on an asynchronous messaging model called Ultra Messaging Persistence. RulePoint uses UM store for communicating with all the processes across the source controller, responder, and event processor.

When application services run on multiple nodes, that is, separate Java virtual machine (JVM) processes, the JVMs communicate over the ultra messaging communications layer. The UM store delivers information, for example, from one source controller to a rule deployed in a separate JVM. Ultra messaging persists information in a series of proprietary files until the messages are acknowledged. If one component is not available, the messaging layer keeps the information until the component recovers.

Configuring a UM Store

RulePoint installs the default topology consisting of a single instance of the UM store. You can have multiple instances of the UM store, and distribute these stores across nodes.

Running multiple instances of umstored allow your applications to continue operation if any one the UM store fails. You also have the option to edit or remove a UM store in a topology.

Adding a UM Store to a Default Topology

The default topology consists of a single instance of the UM store. You can add one or more UM stores to the default topology configuration. Each host can have only one instance of the UM store.

1. Shut down the run-time topology.
2. Add a host to the default topology where you want to add the UM store.
3. On the **Administration** tab, click the **Topology** view.
4. Under **Topology Model** in the navigator, click **UM Store**.
5. From the **Actions** menu in the upper-right pane, select **Add UM Store**.

6. Under **Details**, enter the following configurations:
 - a. Under **Name**, provide a name for the UM store.
 - b. Under **Description**, optionally provide a description for the UM store.
 - c. Under **Host**, enter the IP address of the host where you want to run the UM Store.
 - d. Under **Port**, enter the port number for the UM Store. The port number must be unique. Select a port number between 15000 and 15999.
7. Under **Properties**, enter the value for port number of the HTTP monitor for the UM store.
8. Click **Save**.

The contents panel displays the configured UM store in the default topology. The contents panel header also displays the total number of UM stores configured. The details view displays the details and properties of the UM store.
9. Start the run-time topology.

Editing a UM Store

You can edit the properties of a configured UM store. You can also choose to run the configured UM store on another host.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **UM Store**.
4. Select the UM store that you want to delete, and select **Edit** from the menu.
5. Edit the configurations for the UM store.
6. Click **Save**.
7. Start the run-time topology.

Viewing a Specific UM Store

You can view a specific UM store to view its details and properties.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **UM Store**.
3. In the contents panel, select the UM store that you want to view, and click **Details View**.

The details view displays the details and properties for the selected UM store.

The following table describes the details and the properties for each UM store:

Property	Description
Details	The details of the selected UM store, such as: <ul style="list-style-type: none">- Name. Name of the UM store.- Description. The description of the UM store.- Host. The host IP address.- Port. The port number.- Created By. The user who last created the UM store.- Modified By. The user who last modified the UM store.- Created Date. The date when the UM store was last created.- Last Modified Date. The date when the UM store was last modified.
Properties	The properties of the selected UM store, such as: <ul style="list-style-type: none">- Key. Name of the Java Virtual Machine (JVM) property that runs the java process for the UM store.- Value. The current value of the java property.- Initial Value. The initial value of the java property.

Viewing All UM Stores

You can view all the configured UM stores in the contents panel. You can also sort the UM stores by name or description.

1. On the **Administration** tab, click the **Topology** view.
2. Under **Topology Model** in the navigator, click **UM Store** in the displayed topology default folder.

The contents panel displays the list of UM stores sorted alphabetically by name. You can sort the UM stores by the name or description.

Deleting a UM Store

You can remove a UM store from a topology configuration. When you have only one UM store in a topology, you cannot delete that UM store.

1. Shut down the run-time topology.
2. On the **Administration** tab, click the **Topology** view.
3. Under **Topology Model** in the navigator, click **UM Store**.
4. In the contents panel, select the UM store that you want to delete, and select **Delete** from the menu.
A message prompts you to verify if you want to delete the UM store.
5. Click **OK**.
6. Start the run-time topology.

CHAPTER 9

High Availability

This chapter includes the following topics:

- [High-Availability Overview in RulePoint , 88](#)
- [Achieving High Availability, 89](#)
- [Workflow for Configuring High Availability, 89](#)
- [Introduction to Resiliency in RulePoint, 90](#)
- [Best Practices for Achieving High Availability, 96](#)

High-Availability Overview in RulePoint

You can implement RulePoint to deliver as a high-availability solution to meet your business goals. High availability eliminates single points of failure.

The run-time architecture allows you to create various high-availability designs to improve the RulePoint performance. The run-time processes continue to run despite temporary failures, and ensure minimal service interruption if a service fails.

The following interrelated features in RulePoint maintain end-to-end availability of the run-time services:

- **Resilience.** The ability of RulePoint run time to tolerate temporary connection failures until either the resilience timeout expires or you fix the failure.
- **Restart and failover.** The restart of a service or task after the migration of a service from a primary instance to a backup instance when the primary service instance becomes unavailable.
- **Recovery.** The completion of run-time operations after service interruption. After a service process restarts or fails over, it restores the service state and recovers the run-time operations.

Before you begin your high-availability design, you need to understand your business requirements and which high-availability solution you need to use to solve a particular business problem. You can group the run-time components accordingly to strengthen the high availability in your environment.

Achieving High Availability

In RulePoint, you can create primary and multiple backup instances for the application services, such as the event processor, source controller, and responder controller.

When you configure the application services, you must select the mode as high availability. In a high-availability mode, you need to distribute the primary and backup instances on different nodes. Nodes can be distributed across same or multiple host machines. If one of the service instance becomes unavailable on one node, the grid manager ensures that the secondary service takes over as the primary, without any loss of events.

The primary service instance on a node takes care of all its activities, whereas the backup instance on another node waits passively. When the primary instance goes down, the backup instance becomes the primary and performs the activities of the primary. Meanwhile, the previous primary instance becomes the backup instance. The ultra messaging layer picks up events that occur between the failover from the primary to the backup instance, ensuring that none of the events are lost. In RulePoint, high availability switches over from the primary to the secondary instance seamlessly.

You can also create multiple instances for the grid manager. Each host can have only one grid manager. When you start RulePoint in a particular host, the grid manager in that host is designated as the leader, and the secondary instance is designated as the backup. Only the primary instance can be active at a time.

You can also create multiple instances for the UM store and UM lbmrd for scalability. Each host can have only one instance of UM lbmrd and UM store.

The grid manager controls the primary and the backup run-time components, and manages the change of state from primary to secondary. In a failover situation, the grid manager detects a problem with the primary component, and makes the backup the primary.

Workflow for Configuring High Availability

You must start the high-availability configurations from a default topology. You can configure multiple backup instances for the source controllers and event processors. You need to assign these instances to run on different nodes.

When the primary instance goes down, the backup instance becomes the primary and processes the events and rules. Meanwhile, the previous primary becomes the backup instance. A high-availability-enabled mode prevents loss of events.

1. Shut down the run time.
2. On the **Administration** tab, select the default topology, and add one or more nodes.
3. Select the host for each node and provide the port number.
Note: The port number must be unique.
4. To create one or more instances of the run-time component, select the specific run-time component from the Topology tree in the navigator.
5. Add the component to the topology from the **Actions** menu on the upper-right pane.
6. Select the **Deployment Mode** as **High Availability**.
7. Select the node where you want to run the primary instance and the node that you want to run the backup instance, and save the configuration.

Note: You can choose to run multiple backup instances of the application services across nodes.

8. Start the run time.
9. The grid manager initializes the primary and secondary processes, as designated.

Introduction to Resiliency in RulePoint

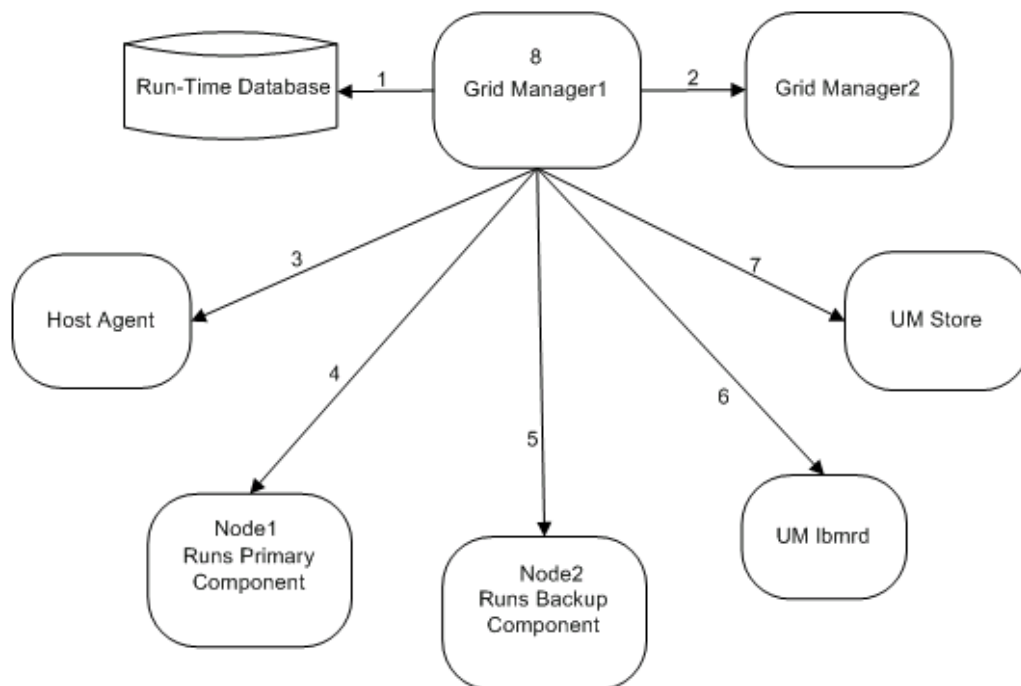
The run-time framework in RulePoint is resilient, such that the run-time components recover quickly, even when there is a failure of one or more of its components.

The grid manager has a built-in emergency email alerting mechanism to alert system administrators of any critical or non-recoverable failure. You need to configure email alerts during installation time.

Failure Scenarios and Failover Actions for the Run-Time Components

During a failover, the grid manager takes action and provides end-to-end availability to maintain business resilience.

The following figure shows the failover scenarios:



The following table describes the actions taken by the failover system in various failure scenarios:

Failover Scenario	Action
1. Database is down.	The grid manager immediately sends an emergency email to the administrator. Meanwhile, the grid manager continues to ping the database until it comes up again. If the database does not come up, the operational state of the system is low. When the database comes up, all the available grid managers and node processes start functioning as usual.
2. Grid manager backup is down.	The primary grid manager sends an emergency email to the administrator. The primary grid manager attempts to restart the secondary grid manager. The number of retries are based on the limit that you have configured.
3. Host agent is down.	The grid manager sends an emergency email to the administrator. The grid manager continues to ping the host agent until it revives.
4. Primary service instance is down.	The grid manager sends an emergency email to the administrator. Meanwhile, the grid manager pings the node that contains the primary instance, a number of times, as configured. If it gets no response from the node, the grid manager designates the backup service as the primary. The grid manager attempts to restart the old primary again, but if it cannot contact the primary because of any network failure, the grid manager designates the backup as primary. The grid manager stops pinging the old primary service. The service brings itself down and restarts as a backup service, preventing a split brain condition.
5. Backup service instance is down.	The grid manager sends an emergency email to the administrator. If the primary is still operational, the grid manager attempts to restart the backup instance on the node.
6. Lbmrdr is down.	The host agent pings the lbmrdr and sends the lbmrdr status to the grid manager. If the grid manager detects that the lbmrdr is down, it attempts to restart the lbmrdr. If the lbmrdr is down, the grid manager sends an emergency email to the administrator.
7. UM store is down.	The host agent pings the UM store and sends the UM store status to the grid manager. If the grid manager detects that the UM store is down, it attempts to restart the UM store. If the UM store is down, the grid manager sends an emergency email to the administrator. If the UM store does not come up, the grid manager brings down the system. If the grid manager cannot restart the UM store, the system operation comes down. The grid manager brings down the system.
8. Primary and backup grid manager is down.	If the primary grid manager goes down, the secondary grid manager takes over. If all the grid managers go down, the grid managers cannot ping any of the nodes. Node fencing logic sets in, and the nodes shut down. If the grid manager does not come up again, the system operation goes down.

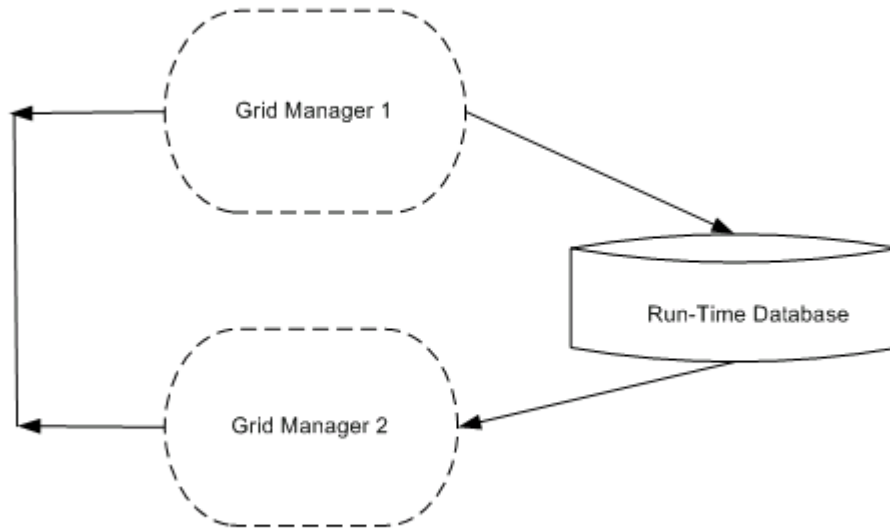
Failover in the Grid Manager

You can have multiple instances of the grid manager to scale the run-time environment. If an instance of the grid manager stops working, the other instance takes over.

The grid manager fails over in the following sequence:

1. When you run the script that starts the RulePoint run-time, the primary grid manager comes up.
2. All topology states persist in the run-time database. The primary grid manager does not have the transient states.
3. The primary grid manager brings up all the configured backups through the host agent.
4. The primary grid manager pings the backups every 10 seconds, as configured.
5. If the backup fails, the primary grid manager brings up the backups again.

The following figure shows the sequence of availability of the grid manager:



Availability in the Grid Manager

You can configure multiple instances of the grid manager to ensure that the run-time environment works efficiently.

Availability in a grid manager occurs in the following sequence:

1. One of the grid manager backups is designated as the primary through leader election by a common lock in the run-time database.
2. The grid manager backups check the run-time database for primary availability as they come up.
3. On detecting that the primary grid manager has failed, the backups trigger leader election.
4. The newly elected primary will try to revive the previously down primary as the backup.
5. Common counter-based lock prevents split-brain condition.

Failover of the Source Controller and the Responder Controller

When you configure service controllers for high availability, no events are lost. The grid manager manages the failover of service controllers.

The source and responder controllers fail over in the following sequence:

1. The grid manager brings up the primary controller and the backups on designated nodes by using the host agents.
2. The grid manager deploys the requests on both the primary and the backup controllers.
3. If the primary controller does not come up, the grid manager designates the backup as the primary. The grid manager attempts to restart the previous primary controller.

4. The grid manager verifies if the new primary controller works fine.

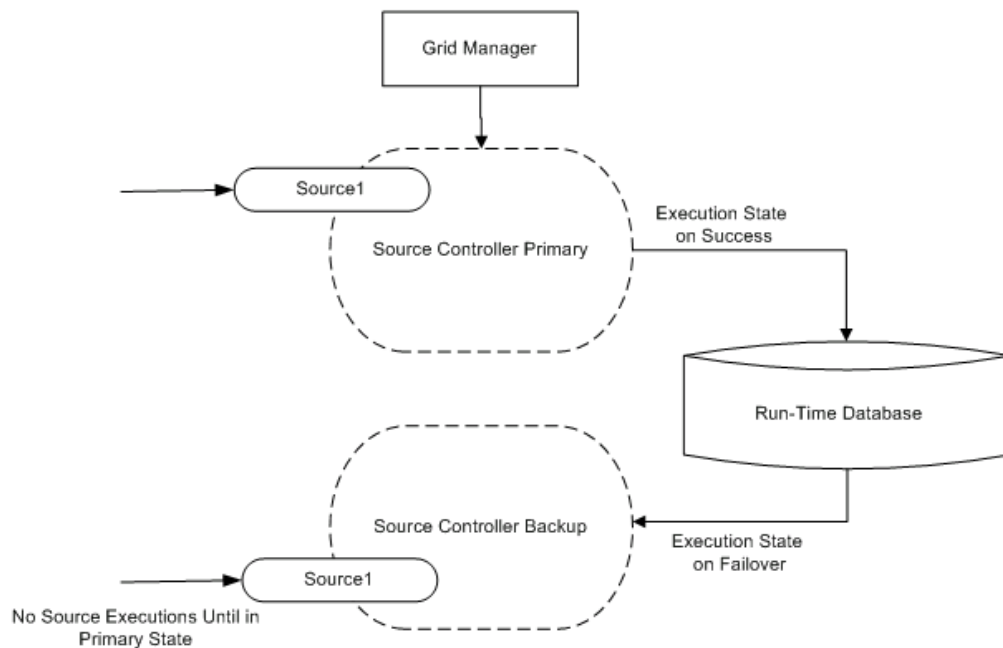
Availability of the Source Controller

If you configure the source controllers for high availability, the primary and backup instances ensure that no events are lost.

- When the primary source controller is up and running, the sources perform the following actions:
 1. The pull sources execute and publish the events.
 2. On successful publishing, the pull sources obtain the stability acknowledgments from the UM store.
 3. The source then persists the source parameters into the run-time database.
- If there is a failure, the sources perform the following actions:
 1. The pull sources of the backup instance start publishing events from where they had stopped by using the run-time database parameters.
 2. Push sources continue to receive messages that are not explicitly acknowledged.

Note: If the primary source controller fails after publishing the events, but before persisting the events into the database, the events might be duplicated.

The following figure shows the sequence of failover in source controllers:

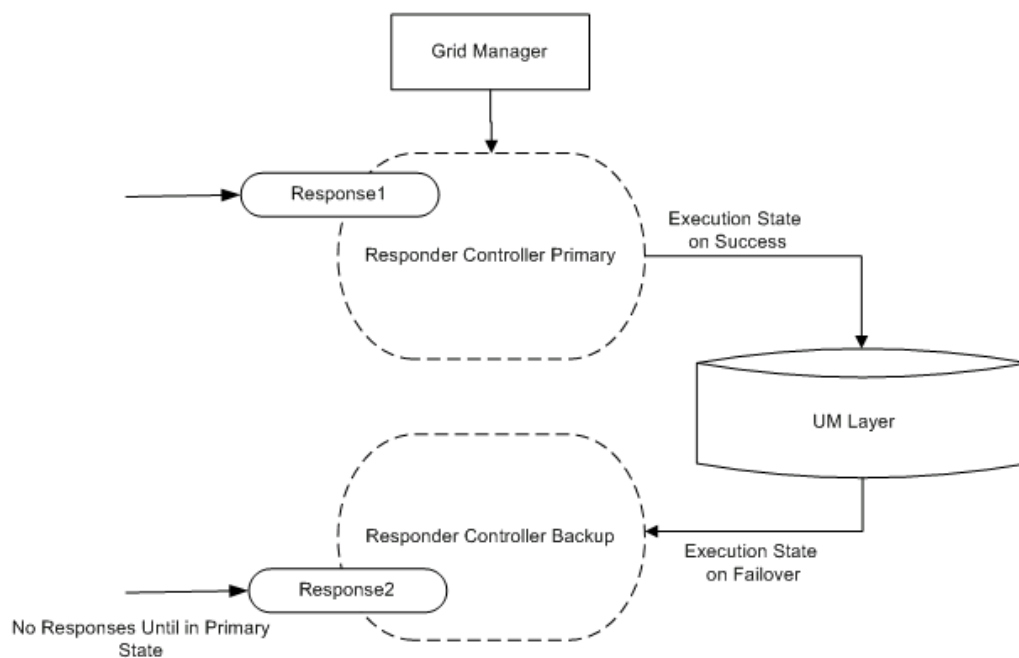


Availability of the Responder Controller

If you configure the responder controllers for high availability, the primary and backup instances ensure that no activations are lost.

- When the primary responder controller is up and running, the responders perform the following actions:
 1. The responders generate responses, and send alerts to the configured external system.

2. For every successful response, the responders acknowledge a message to the UM layer, indicating that it successfully delivered the response to the application.
- If there is a failover, the responders perform the following actions:
 1. The responder controller does a late join, so it does not miss any activations.
 2. The responders in the backup instance start receiving the responses from where they had stopped. The UM layer controls the process by using the explicit acknowledgements that it receives on successful message delivery.
 3. If the primary responder controller fails after generating the responses, but before acknowledging the message to UM layer, the alerts might be duplicated.



When you configure the event processor for high availability, the grid manager manages the failover of the event processor.

- The grid manager uses the host agent to bring up the event processor and the back up on designated nodes.
- If the primary event processor does not come up, the grid manager starts the backup and designates it as the primary.
- The grid manager attempts to start the primary event processor again.
- If all the configured services do not come up, the grid manager does not start.
- The grid manager detects event processor failures through ping and retries, as configured.

- If the grid manager detects that a primary event processor has failed, the grid manager designates the backup event processor as the primary event processor.

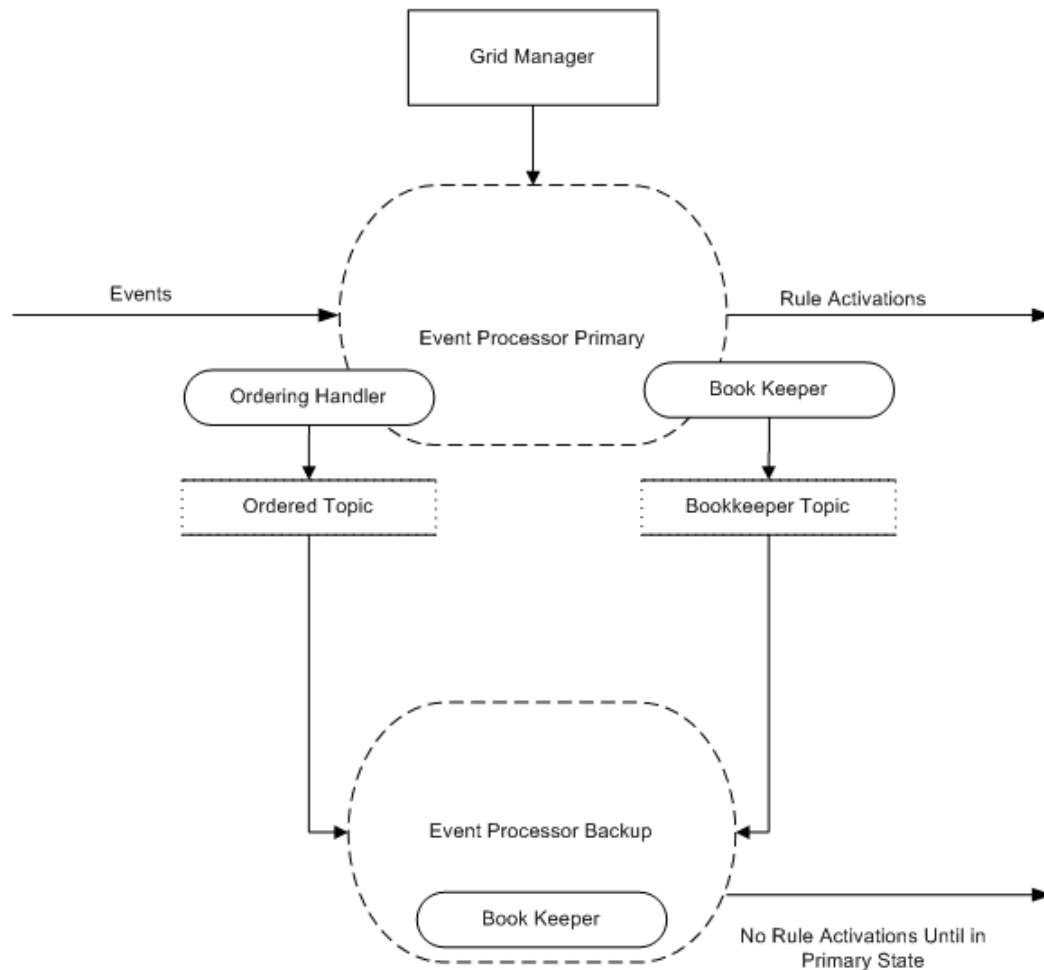
Availability of the Event Processor

When you configure the event processor for high availability, rule processing in the backup event processor is in synchrony with the primary event processor.

The following sequence is involved in an event processor availability:

- The event processor does a late join, so it does not miss any events produced by sources.
- The UM configuration ensures that the event processor sees events in the order that they are produced.
- The ordering handler collects the events from all the topics and publishes it on the ORDERED_TOPIC. As a result, the primary event processor and the backup event processors see the events in the same order.
- When the event processor begins processing, the Book Keeper in the primary event processor publishes the events to the BOOKKEEPER_TOPIC of the backup.
- The backup event processor sees the events from the ORDERED_TOPIC and the BOOKKEEPER_TOPIC. Based on the events on the BOOKKEEPER_TOPIC, it either keeps or clears events received on the ORDERED_TOPIC. The model ensures that the backup is synchronous with the processing activity in the primary event processor.
- The backup is in a continuous warm state, but does not process any rules or produce any activation.
- When a primary event processor is down, the grid manager detects and assigns the backup event processor as the primary.
- When the backup event processor becomes the primary event processor, it initiates its transition from the backup to the primary, and performs the following actions:
 - Attempts to empty the BOOKKEEPER_TOPIC.
 - Blocks receiving events on the ORDERED_TOPIC temporarily.
 - Creates the primary Book Keeper.
 - Feeds the unexpired events in the old Book Keeper for processing.
 - Attaches the primary Book Keeper to get events from the ORDERED_TOPIC.
 - Unblocks receiving events on the ORDERED_TOPIC.

The following figure shows the sequence of events during a failover in an event processor:



Best Practices for Achieving High Availability

To optimize RulePoint availability, ensure that you adopt the best high-availability strategy to configure all the run-time components.

Use the following rules and guidelines when you set up high availability:

- Replicate the application services across multiple nodes. Running the processes on multiple nodes balances the run-time services workload, and processes faster.
- Run the primary instance and the backup instance of the application services on different nodes to ensure that even if one node goes down, another node continues to provide the service. Running the nodes on different host machines ensures that even if one host goes down, the nodes on other hosts will take over the activity.
- Run the active grid manager instance and its backup instance on different hosts, to ensure that even if one host goes down, the grid manager process is still available.
- Run the UM store on different hosts to ensure its high availability.

- Ensure that the database and network are continually available to help in the overall high availability of the system.

CHAPTER 10

Managing Deployment

This chapter includes the following topics:

- [Deployment Overview, 98](#)
- [State of RulePoint Objects, 100](#)
- [Deployment Tasks, 100](#)
- [Deployment Options, 102](#)
- [Deployment Workflow, 102](#)
- [Use Case: Deployment Types and State Transitions, 103](#)
- [Undeploying Running Objects, 104](#)
- [Error Management During Deployment, 105](#)
- [Deployment Considerations and Best Practices, 105](#)
- [Deploying RulePoint Objects, 106](#)
- [Deployment Policy for Templates, 109](#)
- [Redeploying the RulePoint Objects , 110](#)
- [Undeploying the RulePoint Objects, 113](#)
- [Reassigning the RulePoint Objects, 115](#)

Deployment Overview

Deployment involves creating objects in design time and deploying these objects into the application services in run time, so that the configured application services begin processing events.

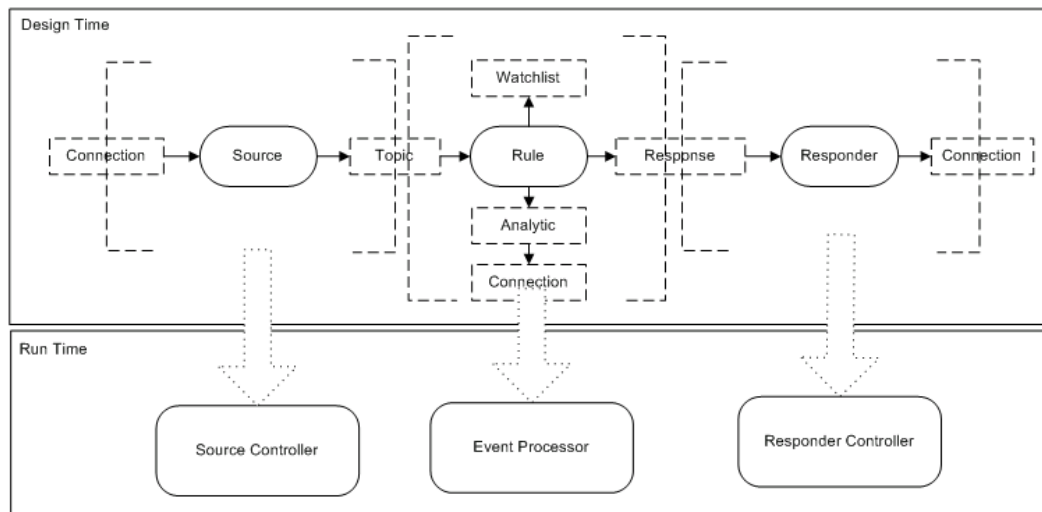
After you complete configuring all the RulePoint objects, you need to deploy the objects from the design-time to the run-time environment. RulePoint objects are of two types, primary objects and supporting objects. Primary objects, which include the source, responder, and rule, drive the execution of the run-time components. The supporting objects, which support the primary objects, include the topics, connections, analytics, watchlists, templates, and responses.

The following table shows the supporting objects for each primary object:

Primary objects	Supporting objects
Source	<ul style="list-style-type: none"> - Topic - Connection
Responder	<ul style="list-style-type: none"> - Response - Connection
Rule	<ul style="list-style-type: none"> - Topic - Analytic - Watchlist - Response - Template

When you deploy the objects, the grid manager deploys the sources into the source controller, responders into the responder controller, while rules into the event processor. You must deploy only the primary objects. The grid manager deploys the supporting objects along with the primary objects. When you try to deploy an object, it will be evaluated and checked for validity. You can deploy the primary object only if it is valid. After you deploy a primary object, the state of that object and its supporting objects changes from Draft to Deployed state. You can choose to deploy the source, rules, and responders in bulk, or you can deploy each object individually.

The following figure shows the deployment of primary and supporting objects into the run-time environment:



You can view the current status of the object, such as Draft, Deployed, or Needs_Deployment on the user interface.

Deployment and Scalability

When you plan to scale your system to handle more events or rules, or increase the available memory of each application service, you need to configure more application services in your run-time environment. Add multiple instances of service controllers and event processors and distribute the deployment of rules, sources, and responders across these instances. If you configure a high-availability environment, you can deploy objects into primary and backup instances of the service controllers and event processors.

After successfully deploying objects from the design time to the run time, source controllers begin to fetch events, rule processors process events, and responder controllers dispatch alerts.

State of RulePoint Objects

State denotes the current status of the RulePoint objects.

The RulePoint objects can have one of the following states:

Draft

When you create and save an object, it is in Draft state. The object is in Draft state when it is created, never deployed, or it has been undeployed. You must deploy the object into the run time to process the objects. An object might be valid or not valid only in the design-time environment. You can move only a valid object to Deployed state.

Deployed

After you create and save a primary object, the object is saved as draft in design time. You must deploy the object to run in the run-time environment. When you deploy a primary object, all supporting objects linked to it are deployed automatically. For example, when you deploy a source, the topic and the connection associated with the source are also deployed. You can deploy an object only when the object is in Draft state. An object is valid both in the runtime and design time environment. When a primary object and its supporting objects are deployed successfully, the object is in Deployed state.

You can undeploy an object if you need to change the object properties. For example, for a JDBC connection, if you want to change the database connection information, you can undeploy the associated source. You can undeploy objects only when it is in deployed state. When you undeploy a primary object, all secondary objects associated with it are also undeployed. After successful undeploy, the state of the object changes to draft. After you complete the changes, you can deploy the objects again.

When you edit a primary object, you can choose to save it first as draft and later deploy it. You can also choose to save and deploy simultaneously by using the save and deploy option.

Needs_Deployment

You can edit or update the deployed primary or the associated secondary objects without performing an undeploy. When you edit and save the objects, the state of the objects changes to Needs_Deployment state. You can edit a deployed secondary object. If the secondary object has an associated primary object, you can use the save and update option so that all the associated primary objects are also deployed.

You can redeploy the objects that you have changed without performing an undeploy. In this case, the changes are affected only when you redeploy the objects. Objects that need redeploy are in Needs_Deployment state.

You can map deployed objects in one service controller to another set of objects in another service controller. You can map or reassign objects from one service controller to another only when the objects are in the Needs_Deployment state. On successful completion of reassign, the state of the objects changes to deployed.

Deployment Tasks

Deployment activities involve deploy, undeploy, redeploy, and reassign tasks. If you have a template, you need to create a deployment policy. You must have execute permissions to perform the deployment tasks.

You can perform the following deployment-related tasks for sources, rules, and responders from the user interface:

Deploy

Use the deploy option to deploy RulePoint objects, such as the source, rule, or responder from the design time to the corresponding application services in the run time. When you deploy, if you have configured multiple application services, you must map the objects to respective application services. You cannot delete any object that is in the deployed state.

Undeploy

Use the undeploy option to undeploy any deployed object. If you want to delete any object that is already deployed, you must first undeploy the object. You can also remove the object from the run time, and place the object in the design time by undeploying the object. When a primary object is undeployed, its supporting objects are also undeployed if they are not referred in any other deployed primary object.

Redeploy

If you edit a deployed object, you use the redeploy option to redeploy the object. Even if you update any of the deployed supporting objects, their state and the primary objects that refer to it changes to Needs_Deployment. In this case, you must redeploy the primary object.

Reassign

Use the reassign option to reassign mapping of deployed source, rules, and responders from one source controller, event processor, and responder controller to another source controller, event processor, and responder controller. You can also reassign objects from standalone mode to high availability mode, or back to standalone mode. If you have configured the service controller for high availability, the grid manager reassigns the objects to all the configured backup service instances. You can reassign mapped objects only when the objects are in Deployed or Needs_Deployment state. On successful completion of reassign, the state of the objects changes to Deployed.

Other Deployment-Related Tasks

In addition to the listed deployment tasks, you need to configure the following:

Create a deployment policy for templates

If you use templates to create customized rules, you need to create a deployment policy to map the template rules to available event processor instances. The grid manager deploys the rules to the event processor instances that you specify. If you mapped the template to multiple event processor instances, the grid manager deploys the rules across the configured event processors in a round-robin fashion. You can also edit or delete a deployment policy.

Configure privileges and permissions for deployment

Configure execute permissions for a user to deploy, undeploy, redeploy, and reassign an object.

View the deployment status, run-time statistics, lineage, and deployment history indicators of the object

You can view the state of the objects in the contents panel. The state for each object can be Draft, Deployed, or Needs_Deployment. The dashboard displays the name, type, and activation count for all deployed objects. It also displays the name, state, and last modified date of all the deployed objects. The related objects view displays the objects.

Validate an object

Verify if the primary and dependent objects are valid for deployment. If the validity holds true for an object, you can perform the deployment-related tasks. If the validity is false, the object is referenced in other objects. In this case, you must resolve the validity to true so that you can carry out the deployment-related tasks.

Update runtime

If you edit a secondary object that is deployed, you can choose to save and update the primary objects that reference the edited secondary object. You can also use the **Update RunTime** feature to redeploy the primary objects that reference it.

Deployment Options

Deployment options include deploy, redeploy, undeploy, and reassign tasks.

The following table shows the possible options that you can use to perform the various deployment-related tasks:

Actions	Options
Deploy an object.	<ul style="list-style-type: none">- Deploy a single source, or multiple sources.- Deploy a single rule, or multiple rules.- Deploy a single responder, or multiple responders.- Deploy multiple sources, rules, and responders simultaneously.
Redeploy an object.	<ul style="list-style-type: none">- Redeploy a single source, or multiple sources.- Redeploy a single rule, or multiple rules.- Redeploy a single responder, or multiple responders.- Redeploy multiple sources, rules, and responders simultaneously.
Undeploy an object.	<ul style="list-style-type: none">- Undeploy a single source, or multiple sources.- Undeploy a single rule, or multiple rules.- Undeploy a single responder, or multiple responders.- Undeploy multiple sources, rules, and responders simultaneously.
Reassign an object.	<ul style="list-style-type: none">- Reassign a single source, or multiple sources.- Reassign a single rule, or multiple rules.- Reassign a single responder, or multiple responders.- Reassign multiple sources, rules, and responders simultaneously.
Create a deployment policy for a template.	<ul style="list-style-type: none">- Create a deployment policy for a template.- Create a deployment policy for multiple template.- Update a deployment policy.- Delete a deployment policy.

Deployment Workflow

You must create the RulePoint objects by using the options in the user interface and deploy the objects from the design time to the run time. You can save the objects in the design time for later deployment.

The following high-level steps guide your through the deployment process:

1. Select the source, rules, and responders that you want to deploy into the run time.
If rules are referenced in templates, you must create a deployment policy for templates.
2. When you deploy, consider the following factors:

- If the run-time environment consists of a single source controller, event processor, and responder controller, you do not have to map the objects to the application services. By default, sources are deployed in the source controller, rules in the event processor, and responders into the responder controller.
- If the deployment environment consists of multiple instances for each type of application service, you must map the sources, rules, responders to the specific application services.

The grid manager deploys RulePoint objects based on the configurations. After deploying all objects, the status of objects changes from Draft to Deployed.

3. After you deploy objects, you can perform the following tasks:

- a. Update or edit the properties of a deployed object.

The state of the updated object and its associated objects changes from Deployed to Needs_Deployment.

Note: If you update a secondary object, and if that object is referenced in other objects, such as templates or rules, the validity of the rule changes from true to false. You need to update the rule as well. Only when the validity of the concerned objects is true, you can redeploy the objects into the run time.

- b. If you have updated an object, redeploy that object.

The state of the object changes from Needs_Deployment to Deployed.

- c. Undeploy a deployed object from the run time to the design time.

The state of the object changes from Deployed to Draft.

Note: If you want to delete an object, you must first undeploy that object, and then delete the object.

- d. If you want to balance the load when adding more rules, sources, and responders, you can also reassign deployed objects across different application services.

If the deployment is successful, the dashboard displays the run-time activity of objects for the deployed components. If the deployment of an object fails at any time during the process, the grid manager rolls back the deployment changes and brings back the system to a consistent state.

Use Case: Deployment Types and State Transitions

Create primary objects, such as the source, rule, and responder, and secondary objects, such as topic, connection, and response. The state of the objects are now in the Draft state.

Deploy the primary objects into the run time. The grid manager also deploys the supporting objects into the run time. The state of the primary and the secondary objects changes from Draft to Deployed.

Scenario 1

- Edit a deployed primary object, for example, change the condition in the query string configuration of the source.
The source changes from the Deployed to the Needs_Deployment state. As the source can use only one topic at a time, the query does not affect any secondary object. The validity of the source remains true.
- Redeploy the source from the design time to the run time.

Scenario 2

- Edit a secondary object, for example, a topic.
Sources and rules use a topic. When you change the name of a topic, a dependent objects dialog box lists all the dependencies of the topic. It lists the rules and sources that conflict with this change. The state of the sources and the rules changes from Deployed to Needs_Deployment. As the source can use only one topic at a time, the validity of the source remains true. However, the validity of the rules changes to false.
- If the validity of the rules is false, edit each of the rules that refer to the topic, to resolve the status.
- Redeploy the sources and rules. You can also use the save and update option, or the update runtime option to redeploy the primary objects that reference the topic.

Scenario 3

Perform the following tasks, based on your requirement:

- If you want to delete a source that you have deployed, you need to undeploy the source first and then delete it.
- If you want to place the deployed source from the run time to the design time, undeploy the source.
When a primary object is undeployed, its supporting objects are also undeployed if they are not referred in any other deployed primary object. The state of the object changes from Deployed to Draft.

Scenario 4

If you have deployed many sources, rules, and responders into the design time, and if there is an overload in the application services, you can reassign the sources, rules, and responders across specific application service instances for load balancing.

Scenario 5

If you want to edit the DRQL statement in a template that is deployed, you can upgrade the template. The dependent objects dialog box lists all the rules created from this template. You can save and redeploy the rules to deploy the associated template rule. The associated template rule changes from Needs_Deployment to the Deployed state.

Undeploying Running Objects

You can select any of the deployed running objects and choose to undeploy. The dependent objects are also undeployed along with the selected object. The grid manager undeploys the objects in the following order:

Sources > Rules > Analytics > Responders

When you try to undeploy running objects, the grid manager undeploys the objects in the following sequence:

1. The grid manager checks its persistent storage to ensure that the service is processing the running objects.
2. The grid manager then starts undeploying the object from the service. The grid manager sends a stop message to all the controller instances where the objects are running. The objects stops consuming events from the queue, clears the backlog, and releases all the resources that it holds.
3. After all the objects respond successfully to the stop message, the grid manager undeploys the object from the service instances.
4. The node removes the transport queue if the object is the last one that is producing events to it.
5. The grid manager then repeats the same steps to undeploy the processes in other nodes.
6. Successful undeployment removes the objects from the persistent repository and reports a success.

7. If there are any errors while undeploying an object, the grid manager retries the operation for the configured number of times before starting the rollback process.
8. As part of the rollback, the grid manager redeploys the undeployed objects with the objects stored in its persistent storage.
9. After redeployment, the grid manager returns an error back to the user interface to notify about the failure.

Error Management During Deployment

The grid manager is responsible for handling errors while deploying the objects. It is imperative that you configure a backup instance for the grid manager, so that if the active grid manager fails, the passive process handles the deployment process.

If there are any errors while you deploy an object, the grid manager retries the operation for the configured number of times. If the retries fail, the grid manager takes one of the following actions:

- If there is a communication error, the grid manager assumes that the process has gone down. The grid manager makes one of the secondary controller instances the primary and then attempts to deploy the object in the new controller instance.
- If there is a deployment error, the grid manager assumes that the deployment cannot proceed and starts the rollback process for undeploying the deployed objects. As part of the rollback process, it proceeds to undeploy all the objects that it had deployed previously on the controller instances. After the rollback is complete, the server returns an error back to the user interface to notify about the failed deployment.
- If the active grid manager process fails while deploying the objects, the passive grid manager process takes over from where the active grid manager process stopped during the deployment process and rolls back the changes.
- If the run time is down during deployment or undeployment, an error message appears, stating that the grid manager cannot be reached.

Deployment Considerations and Best Practices

When you plan to deploy objects from the design time to the run-time environment, consider the following best practices:

- You can distribute execute permissions for deployment and related functions across multiple users.
- You can deploy a primary object only when it is valid. A primary object is valid in the design time when all its supporting objects are valid and resolved.
- When you delete a project, you must undeploy all primary and supporting objects from the run time.
- When you change a primary object, you can deploy it again only if it is valid.
- If you update any supporting object, you need to deploy the primary object again.
- When you undeploy a primary object, all supporting objects that are referred to in the deployed primary object is also undeployed.
- Decide how much the application service can handle. If you need to process more events, you need to configure more application services. You must add multiple instances of service controllers and event processors and distribute the load across specific services.

Deploying RulePoint Objects

You must deploy RulePoint objects in the run-time environment to run the configured services.

You can create the RulePoint objects in the user interface and save it for later deployment. You can group the objects and deploy them in chunks, or deploy all the objects together. For example, you can combine a source, rule, and responder and deploy the objects as a unit or you can deploy an entire set of objects together.

In a default topology, when you deploy objects, you do not have to map objects to the application services. The grid manager deploys sources in the default source controller, responders in the responder controller, and rules in the event processor. If you plan to configure RulePoint for scalability, you can create multiple services and deploy the objects across the service instances. You can also map the objects to application services that are configured for high availability.

If you configure multiple service controllers and event processors, you have the option to map the object to a specific service controller or event processor. During deployment, the grid manager packages all the objects together along with their dependencies and deploys objects into the respective services in the run time. The grid manager opens the deployment package, retrieves each of the configuration details, and proceeds to deploy the objects into the respective service instances. The grid manager deploys the objects in an orderly manner such that it deploys all the dependent objects first, followed by the primary objects.

Prerequisites for Deploying Objects

Before you deploy, you must perform the prerequisite tasks for deploying sources, rules, and responders.

Sources

Complete the following prerequisites before deploying a source:

- Create a topic and a connection for the source.
- Create a source, and a schedule for running the source. The source must be in the Draft state. For instructions, see the *RulePoint User Guide*.
- To scale the solution, create one or more source controllers where you want to deploy the source.

Rules

Complete the following prerequisites before deploying a rule:

- Create a rule. The rule must be in the Draft state.
- Create the related responses, topics, analytics, and watchlists for the rule. For instructions, see the *RulePoint User Guide*.
- If you are referencing rules in a template, you must first create a deployment policy for the template, and then follow the steps for deploying the rule.
- To scale the solution, create one or more event processors where you want to deploy the rule.

Responders

Complete the following prerequisites before deploying a responder:

- Create a response and a connection associated with the responder.
- Create a responder. The responder must be in the Draft state. For instructions, see the *RulePoint User Guide*.
- To scale the solution, create one or more responder controllers where you want to deploy the responder.

Deploying a Single Source

In a default topology, when you deploy the source, it is deployed in the default source controller. To scale your solution, you can create multiple source controllers and distribute the sources across these instances. You can map the sources to source controllers configured for standalone or high-availability mode.

1. On the **Design** tab, click the **Sources** view.
2. In the **Source** contents panel, select the source you want to deploy.
3. Select **Deploy** from the menu, and perform one of the following tasks:
 - a. If the topology consists of only one source controller, when a message prompts you if you want to deploy, click **OK**.
 - b. If you have configured multiple source controllers, either in standalone or a high-availability mode, in the **Deploy Source** dialog box, select the source controller to map the source.
4. Click **Deploy**.

A message appears stating successful deployment and prompts if you want to preview the events.
5. Click **OK**.

A message appears if the events are not yet generated and the source is still executing it.
6. Click **OK**.

The event preview screen displays with all the details of the generated events. You can search for specific events. You have the option to switch on **Auto Refresh** to allow events to display as they enter the RulePoint system.

When you deploy a source, the grid manager deploys the topic and the connection associated with the source. The state of the source and its supporting objects in the Source contents panel changes to Deployed.

Deploying Multiple Sources

You can deploy multiple sources to one or more configured source controllers.

1. On the **Design** tab, click the **Sources** view.
2. In the navigator **Actions** menu, click **Deploy > Sources**
3. Select the sources that you want to deploy, and click **Next**.
4. Select the source controller to map each of the sources.
5. Click **Deploy**.

When you deploy the sources, the grid manager also deploys the topics and connections associated with the source. The state of the source and its supporting objects in the contents panel changes to Deployed.

Deploying a Rule

In a default topology, when you deploy a rule, it is deployed in the default event processor. You can create multiple event processors and distribute the rules across the event processor instances. You can also map the rules to source controllers configured for high availability. When you deploy a rule, the state of the rule and its supporting objects changes from Draft to Deployed.

1. On the **Design** tab, click the **Rules** view.
2. In the contents panel, select the rule that you want to deploy, and click **Deploy** from the menu.
3. Perform one of the following tasks:

- If the topology consists of only one event processor, when a message appears that prompts you if you want to deploy, click **OK**.
- If the topology consists of more than one configured event processor, map the rules to the event processors, and click **Deploy**.

A message appears indicating successful deployment and prompts you if you want to preview the rule trace. You can choose to preview the rule trace or cancel the preview.

Deploying Multiple Rules

You can deploy multiple rules to one or more configured event processors.

1. On the **Design** tab, click the **Rules** view.
2. In the navigator **Actions** menu, click **Deploy > Rules**
3. Select the wizard and advanced rules that you want to deploy, and click **Next**.
4. Select the event processor to map each of the rules, and click **Next**.
5. Select the template rules that you want to deploy, and click **Deploy**.

The state of the rule and its supporting objects changes to Deployed.

Deploying a Single Responder

In a default topology, when you deploy the responder, it is deployed in the default responder controller. You can create multiple responder controllers and distribute the responders across the responder controller instances. You can also map the responders to responder controllers configured for high availability.

1. On the **Design** tab, click the **Responder** view.
2. In the **Responder** contents panel, select the responder you want to deploy, and click **Deploy** from the menu.
3. Perform one of the following tasks:
 - a. If the topology consists of only one responder controller, a message prompts you if you want to deploy, click **OK**.
 - b. If you configured multiple responder controllers, either in standalone or a high-availability environment, in the **Deploy Responder** dialog box, select the responder controller to map the responder.
4. Click **Deploy**.

When you deploy a responder, the grid manager also deploys the supporting objects associated with the responder. The state of the responder and its supporting objects in the Responder contents panel changes to Deployed.

Deploying Multiple Responders

You can deploy multiple responders to one or more configured responder controllers.

1. On the **Design** tab, click the **Responders** view.
2. In the navigator **Actions** menu, click **Deploy > Responders**
3. Select the responders that you want to deploy, and click **Next**.
4. Select the responder controller to map each of the responders.
5. Click **Deploy**.

When you deploy the responders, the grid manager also deploys the objects associated with the responder. The state of the responder and its supporting objects in the responder contents panel changes to Deployed.

Deploying All Objects Simultaneously

After you create all the rules, sources, and the responders, you have the option to deploy all the objects simultaneously. When you deploy, you must map sources, rules, and responders to corresponding instances of the application services. You can deploy objects that are in the draft state.

1. On the **Design** tab, click the **Projects** view.
2. From the navigator **Actions** menu, click **Deploy > Rules, Sources & Responders**.
3. In the **Deploy Rules, Sources & Responders** , dialog box, perform the following tasks:
 - a. Select the advanced and wizard rules that you want to deploy, and click **Next**.
 - b. Select the event processor for deploying the rule, and click **Next**.
 - c. Select the template rules that you want to deploy, and click **Next**.
 - d. Select the sources that you want to deploy, and click **Next**.
 - e. Select the source controller for deploying the source, and click **Next**.
 - f. Select the responders that you want to deploy, and click **Next**.
 - g. Select the responder controller for deploying the responder, and click **Next**.
 - h. Select the responders that you want to deploy, and click **Next**.
4. Click **Deploy**

Deployment Policy for Templates

If you create customized rules by using templates, you need to map the template rules derived from a rule template on available event processor instances from the user interface.

The grid manager deploys the rules to the event processor instances that you specify. If you mapped the template to multiple event processor instances, the grid manager deploys the rules across the configured event processors in a round-robin fashion.

Create a Deployment Policy for a Template

You must create a template rule before deploying a template.

1. On the **Design** tab, click the **Templates** view.
2. In the contents panel, select the template that you want to deploy, and click **Create Deployment Policy** from the menu.
3. Map the template to one or more engine instances where you want to deploy the template.
4. Click **Save**.

Creating a Deployment Policy for Multiple Templates

You have the option to create deployment policies for mapping multiple templates to event engines at a time.

1. On the **Design** tab, click the **Templates** view.
2. In the navigator **Actions** menu, click **Deployment Policies > Create Deployment Policies**.
3. Select the template instances, and click **Next**.
4. Map the templates to one or more engine instances where you want to deploy the template.
5. Click **Save**.

Update a Deployment Policy

1. On the **Design** tab, click the **Templates** view.
2. In the contents panel, select the template for which you want to update the deployment policy, and click **Update Deployment Policy** from the menu.
3. Change the mapping of the selected engine instances to the template.
4. Click **Save**.

Delete a Deployment Policy

You can delete a deployment policy from a project.

1. On the **Design** tab, click the **Templates** view.
2. In the contents panel, select the template for which you want to delete the deployment policy, and click **Delete Deployment Policy** from the menu.

A message appears that prompts you if you want to delete the deployment policy.

3. Click **OK**.

Redeploying the RulePoint Objects

When you edit or update any of the deployed supporting objects, the state of the supporting object and the primary objects that refer to the supporting object change to `Needs_Deployment`. In this case, you must redeploy the primary object.

For example, if you edit a deployed response, the state of the response, the responder that it refers to, and the rule in which the response is used changes to `Needs_Deployment`. The validity of the rule also changes to `false` as it is dependent on the response.

You can redeploy the objects that you have changed without undeploying the objects. In this case, the changes are affected only when you redeploy the objects. Objects that need redeploy are in `Needs_Deployment` state.

Redeploying a Source

After you deploy the source, if you want to change the properties of the source or its supporting objects, you need to redeploy the source. An object must be valid if you want to deploy it to the run-time environment. When you redeploy the source, the state of the source changes to Deployed.

1. On the **Design** tab, click the **Sources** view.
2. In the contents panel, select the source that you want to edit, and click **Edit** from the menu.
3. Edit the details.
The state of the source and all the supporting objects changes from Drafts to Needs_Deployment.
4. Select one the following options:
 - To save and redeploy simultaneously, click **Save and Redeploy**.
 - To save the changes and redeploy later, click **Save**.
 - To redeploy the saved object, click the **Redeploy** button from the menu.

A message appears that prompts you if you want to redeploy the source.

5. Click **OK**.
A message appears stating the redeployment success and prompts you if you want to preview the events.
6. Click **OK**.
The **Event Preview** page appears displaying all the events. You can search for an event from the list.

Redeploying Multiple Sources

You can redeploy multiple sources to one or more configured source controllers. If you make changes to any of the supporting objects, the state of all the primary objects related to it also changes from Deployed to Needs_Deployment. You need to redeploy the primary objects again.

1. On the **Design** tab, click the **Sources** view.
2. In the navigator **Actions** menu, click **Redeploy > Sources**.
3. Select the sources that you want to redeploy.
4. Select the source controller to map each of the sources.
5. Click **Redeploy**.

Redeploying a Rule

After you deploy the rule, if you want to change the properties of the rule or its supporting objects, you need to redeploy the rule. An object must be valid if you want to deploy it to the run-time environment.

1. On the **Design** tab, click the **Rules** view.
2. In the contents panel, select the rule that you want to edit, and click **Edit** from the menu.
3. Edit the rule properties.
If you make changes to any of the supporting objects, a message appears listing the number of dependencies and the details of the conflicting objects.
4. When a message prompts you if you want to continue, click **OK**.
The state of the rule changes from Drafts to Needs_Deployment, and the rule is not valid anymore.

5. To redeploy the edited rule, select the rule in the contents pane, and click **Redeploy** from the menu.
A message prompts you if you want to redeploy the rule.
6. Click **OK**.

Redeploying Multiple Rules

You can redeploy multiple rules to one or more configured event processors.

1. On the **Design** tab, click the **Rules** view.
2. In the navigator **Actions** menu, click **Deploy > Rules**.
3. Select the rules that you want to deploy, and click **Next**.
4. Select the event processor to map each of the rules.
5. If you want to edit the mapping, click **Edit Mapping for Selected Rules**, and remap the event processor instance to the rule.
Click **Save**.

Redeploying a Responder

After you deploy the responder, if you want to change the properties of the responder or its supporting objects, you need to redeploy the responder. An object must be valid if you want to deploy it to the run-time environment.

1. On the **Design** tab, click the **Responders** view.
2. In the contents panel, select the responder that you want to edit, and click **Edit** from menu.
3. Edit the responder properties.
The state of the responder changes from Drafts to Needs_Deployment.
4. To redeploy the edited responder, select the responder in the contents pane, and click the **Redeploy** from the menu.
A message prompts you if you want to redeploy the responder.
5. Click **OK**.

Redeploying Multiple Responders

If you make changes to any of the supporting objects, the state of all the primary objects related to it also changes from Deployed to Needs_Deployment. You need to redeploy the primary objects again.

1. On the **Design** tab, click the **Responders** view.
2. In the navigator **Actions** menu, click **Deploy > Responders**.
3. Select the responders that you want to deploy, and click **Next**.
4. Select the responder controller to map each of the responders.
5. If you want to edit the mapping, click **Edit Mapping for Selected Responders**, and remap the responder controllers to the rule.
6. Click **Save**.

Redeploying All Objects Simultaneously

You must redeploy the primary objects that are in the Needs_Deployment state.

1. On the **Design** tab, click the **Projects** view in the navigator.
2. From the **Actions** menu, click **Redeploy > Rules, Sources & Responders**.
3. Select the rules that you want to redeploy, and click **Next**.
4. Select the sources that you want to redeploy, and click **Next**.
5. Select the responders that you want to redeploy, and click **Redeploy**.

A message appears that indicates successful redeployment of objects.

Undeploying the RulePoint Objects

If you want to delete any object that you have deployed, you must first undeploy the object. You cannot delete any object that is in the deployed state. You can also remove the object from the run time and place it in the design time by undeploying the object.

When a primary object is undeployed, its supporting objects are also undeployed if they are not referred in any other deployed primary object. You can undeploy objects either in bulk or individually.

Undeploying a Source

When you undeploy a source, the state of the source changes from Deployed to Draft.

1. On the **Design** tab, click the **Sources** view.
2. In the contents panel, select the source that you want to undeploy, and click **Undeploy** from the menu.
A message appears that prompts you if you want to undeploy the source.
3. Click **OK**.

Undeploying Multiple Sources

You can undeploy multiple sources at a time.

1. Click the **Design** tab.
2. In the navigator **Actions** menu, click **Undeploy > Sources**.
3. Select the sources that you want to undeploy, and click **Undeploy**.
A message appears that prompts you if you want to undeploy the sources.
4. Click **OK**.

The state of the object changes from Deployed to Draft.

Undeploying a Rule

When you undeploy a rule, the state of the rule changes from Deployed to Draft.

1. On the **Design** tab, click the **Rules** view.

2. In the contents panel, select the rule that you want to undeploy, and click **Undeploy** from the menu.
A message appears that prompts you if you want to undeploy the rule.
3. Click **OK**.

Undeploying Multiple Rules

You can undeploy multiple rules at a time.

1. Click the **Design** tab.
2. In the navigator **Actions** menu, click **Undeploy > Rules**.
3. Select the rules that you want to undeploy, and click **Undeploy**.
A message appears that prompts you if you want to undeploy the rules.
4. Click **OK**.

The state of the rule changes from Deployed to Draft.

Undeploying a Responder

When you undeploy a responder, the state of the responder changes from Deployed to Draft.

1. On the **Design** tab, click the **Responders** view.
2. In the contents panel, select the responder that you want to undeploy, and click **Undeploy** from the menu.
A message appears that prompts you if you want to undeploy the responder.
3. Click **OK**.

Undeploying Multiple Responders

You can undeploy multiple responders at a time.

1. Click the **Design** tab.
2. In the navigator **Actions** menu, click **Undeploy > Responders**.
3. Select the responders that you want to undeploy, and click **Undeploy**.
A message appears that prompts you if you want to undeploy the responders.
4. Click **OK**.

The state of the responder changes from Deployed to Draft.

Undeploying All Objects Simultaneously

If you want to delete any object that you have deployed before, you must first undeploy that object.

1. Click the **Design** tab.
2. From the navigator **Actions** menu, click **Undeploy > Rules, Sources & Responders**.
3. Select the rules that you want to undeploy, and click **Next**.
4. Select the sources that you want to undeploy, and click **Next**.
5. Select the responder controller that you want to undeploy, and click **Undeploy**.

A message appears that indicates successful undeployment of objects.

Reassigning the RulePoint Objects

You can reassign the deployed source, rules, and responders from one source controller, event processor, and responder controller to another source controller, event processor, and responder controller.

For example, if you have deployed a responder in one responder controller, but you want to switch the responder to another responder controller, you can use the reassign option. You can also reassign the responder from standalone mode to high-availability mode. If you have enabled the responder controller for high availability, the grid manager reassigns the sources to all the configured backup controller instances.

You can reassign mapped objects only when the objects are in Deployed or Needs_Deployment state. On successful completion of reassign, the state of the objects changes to Deployed.

Reassigning a Source

You can reassign the mapping of a single source at a time across multiple source controllers.

1. On the **Design** tab, click the **Sources** view.
2. In the contents panel, select the source you want to reassign, and select **Reassign** from the menu.
You can reassign a source if it is in the Deployed state.
3. Select the source controller to which you want to reassign the source.
4. Click **Reassign**.

The grid manager reassigns the responder to the source controller.

Reassigning Multiple Sources

You can reassign the mapping of multiple sources across configured source controllers.

1. Click the **Design** tab.
2. In the navigator **Actions** menu, click **Reassign > Rules**.
3. Select the rules that you want to reassign, and click **Next**.
4. Select the event processor for which you want to reassign the rule, and click **Reassign**.
5. Click **OK**.

Reassigning a Responder

You can reassign the mapping of a responder across configured responder controllers.

1. On the **Design** tab, click the **Responders** view
2. In the contents panel, select the responder you want to reassign, and click **Reassign** from the menu.
You can reassign a responder if it is in the Deployed state.
3. Select the responder controller to which you want to reassign the responder.
4. Click **Reassign**.

The grid manager reassigns the responder to the responder controller.

Reassigning Multiple Responders

You can reassign the mapping of multiple sources across the configured source controllers.

1. Click the **Design** tab.
2. From the navigator **Actions** menu, click **Reassign > Responders**.
3. Select the responders that you want to reassign, and click **Next**.
4. Select the responder controller to reassign the responder, and click **Reassign**.
5. Click **OK**.

Reassigning All Objects Simultaneously

You can reassign mapped sources, rules, and responders simultaneously. Reassign is possible for objects that are in Deployed or Needs_Deployment state.

1. Click the **Design** tab.
2. From the navigator **Actions** menu, click **Reassign > Rules, Sources & Responders**.
3. Select the rules that you want to reassign, and click **Next**.
4. Select the event processor for deploying the reassigned rule, and click **Next**.
5. Select the sources that you want to reassign, and click **Next**.
6. Select the source controller for deploying the reassigned source, and click **Next**.
7. Select the responders that you want to reassign, and click **Next**.
8. Select the responder controller for deploying the reassigned responder, and click **Next**.

A message appears that indicates successful reassignment of objects.

CHAPTER 11

Dashboard

This chapter includes the following topics:

- [Overview of Dashboard, 117](#)
- [Using the Dashboard, 118](#)
- [Objects View for an Application Service, 122](#)
- [Task Types for Objects on the Dashboard, 126](#)
- [Use Case: Understanding Dashboard Functions and Troubleshooting, 129](#)

Overview of Dashboard

The dashboard serves as a reporting module for the run-time processing in RulePoint. It is a real-time user interface, showing a graphical presentation of the run-time status and trends of objects.

The dashboard provides real-time visibility to the RulePoint performance, and helps you make instantaneous decisions. You must use the functions on the dashboard to manage objects.

Dashboard Interface

The RulePoint dashboard is visually intuitive, easy-to-read, and provides a summary of the configured application services.

The dashboard displays the following information for an application service:

- Source controllers, responder controllers, and event processors configured on nodes.
- Availability of the application services on nodes.
- Configured primary and backup nodes in a high-availability setup.
- Objects deployed on source controllers, responder controllers, and event processors on specific nodes.
- CPU and memory consumption on each node.
- Name, type, and activation count for all deployed objects.
- Name, state, and last modified state of all deployed objects.
- Average throughput, maximum latency, and average latency of the deployed sources and responders.
- Number of events in topics, number of evaluations and number of activations for rules, and number of alerts for responses.
- Functions that you can administer, for example, stop events from publishing, resume processing, enable rule tracing, purge events, and troubleshoot events.

- Run-time logs for the application services.
- Summary of generated events, where you have the option to search for specific events.

Using the Dashboard

The **Dashboard** tab consists of the **Default**, **Events**, and **Logs** views.

Metrics View

The **Metrics** view is the default view of the dashboard.

In the **Metrics** view, you can see the following information:

Select Time Line

You need to select the time line to see the metrics of objects in the run time. You can set it anywhere between 5 minutes and 24 hours. For example, if you set the time line at five minutes, you can see the metrics for the last five minutes.

Application Services, System Services, Hosts, and Nodes in a Topology

The left panel displays the name of the topology and the configured application services, such as source controllers, event processors, and responder controllers in that topology. You can also view the hosts, nodes, activity manager, and system services, such as the grid manager, UM lbmrd, or UM store.

When you rest the pointer over each application service, you can view the following information:

- **Name.** Name of the application service. If it is a host, the IP address displays.
- **Status.** The status of a service on the node, whether the application service is running or stopped. Green shows Running status, while Red shows Stopped status.
- **Type.** The type of service, such as application service, system service, node, host, or activity manager.

Select an application service to highlight the node and host which contains that application service. If you configure the application service on a standalone node, it shows one instance of the node. If you configure the application service for high availability, it shows the primary and backup instances of the nodes. The node that consists of the primary instance displays as 1 and the node that consists of the backup instance displays as 2.

You can view the refresh icon in the upper-left panel to refresh the dashboard. You can also refresh the dashboard, purge the metrics of the topology or the controllers from the **Actions** menu.

Host CPU and Memory Utilization

When you select an application service, the lower left panel displays the following views:

- **Details** view. Displays the name of the application service, the type of application service and the status. It also displays the name of the node and host on which you configure the application service.
- **Host CPU/Memory Usage** view. Displays a graphical representation of the % CPU and % memory utilization for the configured application service.

Objects View

When you click an application service on the left panel, you can view the deployed primary and supporting objects for that application service on the right panel.

- Select a source controller to view the deployed sources and topics.
- Select an event processor to view the configured rules, responses, analytics, topics, and watchlists.
- Select a responder controller to view the configured responders and responses.

You can apply filters to drill down to specific information, such as object or project names, type, number of events, alerts, or activations. Use the various options provided to filter items.

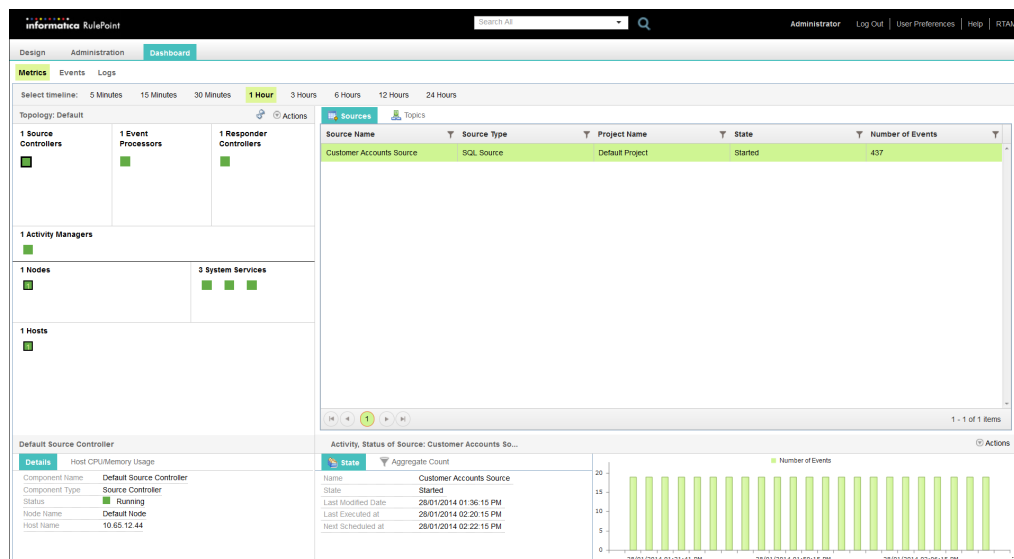
Activity and Status of Objects

When you select an object on the upper-right panel, the lower right panel displays the status and aggregate count, and a graphical view of the activity of the deployed object. You view the events generated for a source controller, activations for an event processor, and alerts for a responder controller. When you rest the pointer on the graph, you can view the number of events, activations, or alerts at a specified time and day.

Note: If you use Internet Explorer 8, you can see an aggregate of events, with a maximum of 30 plottings at a time.

You can use the **Actions** menu displayed on the lower-right panel to perform the tasks related to that object.

The following figure shows the **Metrics** view of the dashboard:



Events View

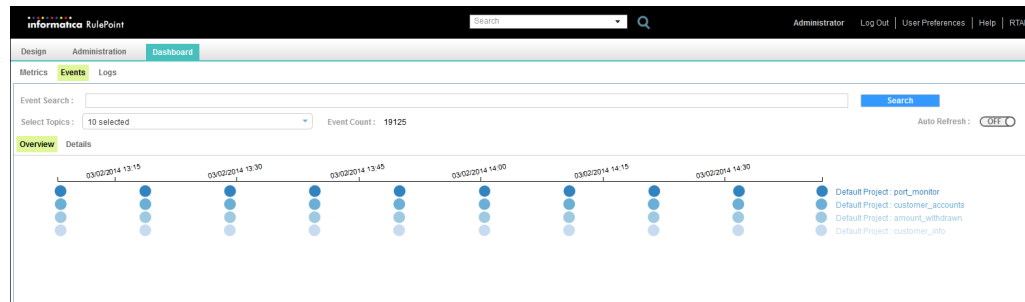
All the events generated using RulePoint are saved in the database. You can choose to view all the generated events or search for specific events. You can also choose to view the events as they enter into the system.

The events view helps you understand the type of events that enter into the system. A review of the events will help you quickly identify patterns in data so that you can build effective rules.

You can view the following features in the **Events** view:

- **Event Search.** To search for events generated for a topic and to view details of the event, you can perform one of the following tasks:
 - Provide the data to search so that the search result displays all the events that contain that data.
 - Select a topic from the **Select Topics** menu and search for an event with the specified data in the search.
 - You can provide any text to search or you could provide the key-value pairs in the search field. The key is the column name and the value is the data in the column. For example, type the column name = value to search for specific events. You can use multiple key-value pairs, separated by AND, OR, or NOT. The search text supports wildcard characters, such as ? and *. The search text must not start with the NOT condition.
- **Event Count.** To view the number of events displayed for the search from the total number of events in the system.
- **Overview** view. Select this view for a display of event data calibrated in graphical format. Data spans the number of events aggregated over a period of time relative to a topic. A circle appears below the scale to depict the number of events for a particular topic. The size of the circle is proportional to the number of events. When you point the mouse over the dot, the event count at a particular time and date for the topic is displayed. You can also view the topic name and the project in which the topic is configured. When you point the mouse over a topic name on the right of the graph, the number of events replace the circles to depict the number of events generated for the selected topic. Double-click the topic name to view the details of the topic.

The following figure shows the **Overview** view within the **Dashboard** tab:



- **Details view.** Select the **Details** view for details of the event for a particular topic in tabular format. The table displays the source name, the timestamp, and the properties of the topic. The following figure shows the **Details** view within the **Dashboard** tab:

The screenshot shows the Informatica RulePoint Dashboard. The top navigation bar includes 'Design', 'Administration', and 'Dashboard' (selected). Below this, there are tabs for 'Metrics', 'Events', and 'Logs'. The 'Events' tab is active, showing a search bar, a dropdown for 'Select Topics' (10 selected), and an 'Event Count' of 18866. An 'Auto Refresh' toggle is set to 'Off'. The main content area displays two tables under the 'Details' view.

Default Project: port_monitor

sourceName	timeStamp	port	message	host	service_id
Monitor Server Port	03/02/2014 14:40:25	8161		imvh58cep53	NOT OK
Monitor Server Port	03/02/2014 14:40:12	8161		imvh58cep53	NOT OK
Monitor Server Port	03/02/2014 14:39:59	8161		imvh58cep53	NOT OK
Monitor Server Port	03/02/2014 14:39:46	8161		imvh58cep53	NOT OK
Monitor Server Port	03/02/2014 14:39:33	8161		imvh58cep53	NOT OK
Monitor Server Port	03/02/2014 14:39:20	8161		imvh58cep53	NOT OK
Monitor Server Port	03/02/2014 14:39:07	8161		imvh58cep53	NOT OK
Monitor Server Port	03/02/2014 14:38:54	8161		imvh58cep53	NOT OK

Default Project: customer_accounts

sourceName	timeStamp	total_bal	street	pin_code	acc_activation_date	cust_id	city	state1	acc_type	acc_no	name	dob	phone_no	email_id	acc_status
Customer Accounts Source	03/02/2014 14:40:27	45000	8100 Sunrise Lakes Blvd.	33322	2009-12-01 00:00:00.0	1004	Sunrise	FL	Savings	97	Dawn Schwartz	1985-06-12 00:00:00.0	9546253319	Dawn.schwartz@gmail.com	Active
Customer Accounts Source	03/02/2014 14:40:27	5000	8100 Sunrise Lakes Blvd.	33322	2010-12-01 00:00:00.0	1004	Sunrise	FL	Demat	48	Dawn Schwartz	1985-06-12 00:00:00.0	9546253319	Dawn.schwartz@gmail.com	Active
Customer Accounts Source	03/02/2014 14:40:27	30500	641 Landau Drive	55125	2009-12-01 00:00:00.0	1006	Palm Bay	FL	Savings	14	Isabel A Quam	1992-07-01 00:00:00.0	9728514685	isabel@hotmail.com	Active

- **Export.** Use the **Export** icon to export the latest 10,000 events generated for a particular topic into a .CSV file.
- **Auto Refresh.** Switch on Auto Refresh if you want the dashboard to display events as they enter the system. By default, **Auto Refresh** is switched off.

Logs View

Use the **Logs** view to see the run-time logs, or to check for errors.

You can view all the run-time logs, with details such as the RulePoint component that is responsible for the log, the node that hosts that component, the log level (INFO, ERROR, or WARN), the component type (node, source controller, event processor, or responder controller), the log message, and the log time.

You can double-click a specific log for more information about logs. If there is an error, use the stack trace to view specific details of the error.

When the log message has an ID, double click that log and then hover over the ID to view the details, such as the name, description, and type of object, and the project that the object belongs to.

You can filter logs based on the node name, log level, component type, log message, and log time. You can choose to filter the logs from the **Log Time** in the following ways:

- View logs generated from the last day or last week.
- Use the custom view to select the start date and end date to view logs generated for that period.
- View logs based on the node name, log level, component type, or log message.

You also have the option to sort the logs based on the log time.

Actions menu: You can select a log, and use the **View Log Details** to view the details for a log. You can also purge the log data. You need to provide the date and time for purging the logs. For example, if you set the date at 2014/01/31 02:05:07.000, the purge action will delete all logs before the specified time and day.

The following figure shows the **Logs** view within the **Dashboard** tab:

Node Name	Log Level	Component Type	Message	Log Time
Default Node	INFO	Node	Total published expiration from rules 0	31/01/2014 10:29:57 AM
Default Node	INFO	Node	Total published expiration from engine 0	31/01/2014 10:29:57 AM
Default Node	INFO	Node	Queued events in primary queue handler 0	31/01/2014 10:29:57 AM
Default Node	INFO	Node	Total published activations 0	31/01/2014 10:29:57 AM
Default Node	INFO	Node	Total published evaluations by rules 0	31/01/2014 10:29:57 AM
Default Node	INFO	Node	Number of events referred by engine 0	31/01/2014 10:29:43 AM
Default Node	INFO	Source Controller	jobWasExecuted() recreate fixed delay trigger: 120000	31/01/2014 10:29:23 AM
Default Node	INFO	Source Controller	Execution of Service : null (id=5825818-3a8f-45e9-bc1b-b8126d048c8) (exec=1391144362990) completed in 0 se...	31/01/2014 10:29:22 AM
Default Node	INFO	Source Controller	SQL Source Query: select * from customer_info c,account_info a where c.cust_id=a.cust_id parameters: []	31/01/2014 10:29:22 AM
Default Node	INFO	Source Controller	jobToBeExecuted() called	31/01/2014 10:29:22 AM
Default Node	INFO	Node	Queued events in primary queue handler 0	31/01/2014 10:28:57 AM
Default Node	INFO	Node	Total published evaluations by rules 0	31/01/2014 10:28:57 AM
Default Node	INFO	Node	Total published expiration from rules 0	31/01/2014 10:28:57 AM
Default Node	INFO	Node	Total published expiration from engine 0	31/01/2014 10:28:57 AM
Default Node	INFO	Node	Total published activations 0	31/01/2014 10:28:57 AM
Default Node	INFO	Node	Number of events referred by engine 0	31/01/2014 10:28:43 AM
Default Node	INFO	Node	Total published expiration from rules 0	31/01/2014 10:27:57 AM
Default Node	INFO	Node	Total published activations 0	31/01/2014 10:27:57 AM
Default Node	INFO	Node	Total published expiration from engine 0	31/01/2014 10:27:57 AM
Default Node	INFO	Node	Total published evaluations by rules 0	31/01/2014 10:27:57 AM
Default Node	INFO	Node	Queued events in primary queue handler 0	31/01/2014 10:27:57 AM
Default Node	INFO	Node	Number of events referred by engine 0	31/01/2014 10:27:43 AM
Default Node	INFO	Source Controller	Execution of Service : null (id=5825818-3a8f-45e9-bc1b-b8126d048c8) (exec=1391144242980) completed in 0 se...	31/01/2014 10:27:22 AM

Objects View for an Application Service

When you select an application service on the left panel, you can view the deployed objects in that application service. The contents panel and the activity and status panel display the deployed objects and the configured information for that object.

Source Controller

When you select a source controller on the left pane of the dashboard, the **Sources** and **Topics** views in the right pane displays the deployed sources and topics in that source controller.

You can also view the node and host where you configured the source controller.

The following table describes the displayed configurations for the **Sources** and **Topics** view for a source controller:

Object Type	Contents Panel	Activity and Status Panel
Sources	<p>Displays the following information:</p> <ul style="list-style-type: none"> - Name of the source - Source type - Project name - State of the object - Number of events for the set time line 	<p>Displays the following activity and status for each source:</p> <ul style="list-style-type: none"> - Under State: <ul style="list-style-type: none"> - Name of the source - Started or stopped status of the source, and errors, if any - Last modified date and time of the source - Last executed date and time - Next scheduled date and time according to the set time line - Under Aggregate Count: <ul style="list-style-type: none"> - Number of activations/second - Average latency/second - Maximum latency/second - A graph on the right displays the number of events for the set time line. - You can also administer the following functions for sources: <ul style="list-style-type: none"> - Purge the source metrics - Run the source once - Start the source - Stop the source
Topics	<p>Displays the following information:</p> <ul style="list-style-type: none"> - Name of topics - Project name - Number of events for each topic for the set time line 	<p>Displays the activity and status for each response:</p> <ul style="list-style-type: none"> - Under State: <ul style="list-style-type: none"> - Name of the topic - Deployed date - Deployment state of the object - Under Aggregate Count, the dashboard displays the number of events for the set time line. - You can use the View Topic function for the following details: <ul style="list-style-type: none"> - Event from source - Source type - Time stamp - You can also administer the following functions for topics: <ul style="list-style-type: none"> - Purge the topic metrics - View a topic - Create an event - Copy an event

Event Processor

When you select an event processor on the dashboard, you can view the deployed objects for the event processor in the contents panel.

The following table describes the displayed configurations for an event processor:

Object Type	Contents Panel	Activity and Status Panel
Rules	Displays the following information: <ul style="list-style-type: none">- Name of the rule- Project name- State of the rule- Number of activations for the selected time line	Displays the following activity and status for each source: <ul style="list-style-type: none">- Under State:<ul style="list-style-type: none">- Name of the rule- Started or stopped status, and errors, if any.- Date and time when the rule was last modified.- Under Aggregate Count:<ul style="list-style-type: none">- Number of activations- Number of evaluations- A graph on the right displays the number of events for the set time line.- You can administer the following functions for rules:<ul style="list-style-type: none">- Purge the rule metrics- Start the rule- Stop the rule- View the trace report- Enable rule tracing- Disable rule tracing
Responses	Displays the following information: <ul style="list-style-type: none">- Name of the response- Type of response- Project name- Number of alerts	Displays the activity and status for each response: <ul style="list-style-type: none">- Under State:<ul style="list-style-type: none">- Name of the response- The deployed date and time- Deployment state of the response- Under Aggregate Count, displays the number of alerts.- You can use the view the response function for the following details:<ul style="list-style-type: none">- Response sent by the responder- Responder type- Time stamp- You can administer the following functions for responses:<ul style="list-style-type: none">- Purge the response metrics- View the responses
Analytics	Displays the following information: <ul style="list-style-type: none">- Name of the analytic- Analytic type- Project name- Number of invocations	Displays the activity and status for each topic: <ul style="list-style-type: none">- Under State:<ul style="list-style-type: none">- Name of the analytic- The deployed date and time- Deployment state of the analytic.- You can administer the following functions for analytics:<ul style="list-style-type: none">- Purge the analytics metrics- View the analytics

Object Type	Contents Panel	Activity and Status Panel
Topics	Displays the following information: <ul style="list-style-type: none"> - Name of the topic - Project name - Number of events 	Displays the following information: <ul style="list-style-type: none"> - Under State: <ul style="list-style-type: none"> - Name of the topic - The deployed date and time - Deployment state of the topic - Under Aggregate Count: <ul style="list-style-type: none"> - Number of events - Create events - You can use the following functions: <ul style="list-style-type: none"> - Purge the topic metrics - View the topic
Watchlists	Displays the following information: <ul style="list-style-type: none"> - Name of the topic - Project name - Element Count 	Displays the following information: <ul style="list-style-type: none"> - Under State: <ul style="list-style-type: none"> - Name of the watchlist - The deployed date and time - Deployment state of the watchlist - Under Aggregate Count, displays the watchlist content - You can administer the following function for watchlists: <ul style="list-style-type: none"> - View the watchlists

Responder Controller

When you select a responder controller on the left pane of the dashboard, the **Responders** and **Responses** views in the right pane display the deployed responders and responses in that responder controller.

The following table describes the displayed configurations for a responder controller:

Object Type	Contents Panel	Activity and Status Panel
Responders	Displays the following information: <ul style="list-style-type: none">- Name of the responder- Responder type- Project name- Responder state- Number of alerts for each responder type	Displays the following activity and status for each source: <ul style="list-style-type: none">- Under State:<ul style="list-style-type: none">- Name of the responder- Started or stopped status, and errors, if any- The date and time when the responder was last modified- Under Aggregate Count:<ul style="list-style-type: none">- Average throughput of events per second- Average latency in seconds.- Maximum latency in seconds.- A graph on the right displays the number of alerts for the set time line.- You can also administer the following functions for responders:<ul style="list-style-type: none">- Purge the responder metrics- Start the responder- Stop the responder
Responses	Displays the following information: <ul style="list-style-type: none">- Name of the response- Type of response- Project name- Number of alerts	Displays the activity and status for each response: <ul style="list-style-type: none">- Under State:<ul style="list-style-type: none">- Name of the response- The deployed date- Deployment state of the response- Under Aggregate Count, displays the number of alerts.- You can also administer the following functions for responders:<ul style="list-style-type: none">- Purge the response metrics- View the responses- You can view the following response details:<ul style="list-style-type: none">- Response sent by the responder- Responder type- Time stamp- Response details

Task Types for Objects on the Dashboard

The **Actions** menu on the lower-right pane of the RulePoint dashboard provides the tasks that you can perform for deployed objects. The options provide an understanding of how RulePoint progresses towards completing a particular task for a deployed object.

Task Types for Objects in Source Controller

The **Actions** menu in the lower-right pane contains the tasks you can perform for objects deployed in a source controller. Select the deployed object in a source controller to perform the tasks.

When you select a source controller on the left pane, you can perform the following tasks for a source deployed on that source controller:

Function	Action
Purge object metrics	Purges all event count details for a source from the dashboard. It refreshes the page after the purge completes.
Run once	Runs the source once when you manually trigger the source. The source polls the data only one time and publishes them as events. You can use this option only for schedulable sources.
Start source	Starts the source to poll data and generate events based on a set schedule. By default, when you deploy the source, it generates events. You can use this function if you have stopped the source from polling data.
Stop source	Stops the source from polling data and generating events.

When you select a source controller on the left pane, you can perform the following tasks for topics:

Function	Action
Purge object metrics	Purges event count details and events for a topic deployed on the source controller from the dashboard.
View topic	After an event is published, you can view the topic details, such as the event from the source, the source type, and the timestamp when the event was created. You can filter events based on the source type, event name, and timestamp.
Copy topic	Copies an existing event details and properties for a topic, if you do not want to create a new topic.
Create event	If you add a property to a topic, you can publish an event for that topic. The contents panel displays the configured number of events.

Task Types for Objects in Event Processor

When you select an event processor on the left pane, you can perform the following tasks for a rule:

Function	Action
Purge object metrics	Purges all rule activation details for a rule deployed on the event processor from the dashboard.
Start rule	Starts the rule execution. The lower panel displays the name and state of the rule. It also displays the aggregate count for the number of events created and the number of events activated after meeting the rule condition.
Stop rule	Stops a rule from execution. The state of the rule changes to stop. You can see the number of events and activations between the start and stop of a rule.

Function	Action
View tracing	Displays the event tracing report summary for a rule. Provides details, such as the event, time, topic, and activations for the rule.
Enable rule tracing	Listens to event details of a rule and appends it to a report. You can troubleshoot errors by using the information in the report.
Disable rule tracing	Disables rule tracing for a rule.

When you select an event processor on the left pane, you can perform the following tasks for a response:

Function	Action
View responses	Displays the details for a particular response, such as the response type sent from the source to an admin, the time stamp, and the property of the response.

When you select an event processor on the left pane, you can perform the following tasks for a topic:

Function	Action
View topic	After an event is published, you can view the topic details, such as the event from the source, the source type, and the timestamp when the event was created.
Copy event	Copies a specific event when you view an event from the event list. You can copy an event when you want to run the rule again for that event.
Create event	Publishes an event for a topic when you change the topic property. The contents panel displays the configured number of events.

Task Types for Objects in Responder Controller

When you select a responder controller on the left pane, you can perform the following tasks for a responder:

Function	Action
Purge object metrics	Purges alert details for a responder from the dashboard.
Start responder	Sends alerts to the external system through the responder. The lower panel displays the name and state of the responder. It also displays the average throughput of events per second and the latency to generate an alert.
Stop responder	Stops a responder from dispatching alerts.

When you select a responder controller on the left pane, you can perform the following tasks for a response:

Function	Action
Purge object metrics	Purges alert details for a response from the dashboard.
View responses	After an alert is generated, you can view the response, the responder type, and the timestamp when the alert was created.

Task Types for Objects at Topology and Component Level

When you select an application service on the left panel of the dashboard, you can perform the following task:

Function	Action
Refresh	Refreshes data for all objects for the selected application service.

You can also perform the following tasks from the **Actions** menu in the upper-left pane:

Function	Action
Purge topology metrics	Purges details of all objects across all the configured application services in that topology.
Purge controller metrics	Purges details of all objects within the selected application service.

Use Case: Understanding Dashboard Functions and Troubleshooting

After you deploy the primary and secondary objects into the application services, use the dashboard to view and understand the functioning of all deployed objects, react to real-time running events, and analyze their impact.

You can use the functions in the dashboard to view if the objects are running properly. For example, you need to understand if a source produces events, rules evaluate events as per the configured condition, and responses trigger upon rule evaluation. You can also understand what errors are encountered at each point that will help you troubleshoot appropriately.

The use cases will help you understand some of the tasks that you can perform on the dashboard.

Scenario 1: Viewing Topics and Events for a Source

1. Click the **Dashboard** tab.
2. In the topology view, select the time line for which you want the statistics for deployed objects.
3. To view the statistics for a specific source, select the source controller where you deployed the source.

The contents panel displays the sources, topics, and connections deployed in that source controller. It also displays the number of events generated for the source and its properties. The lower panel displays the state, aggregate count, and a graph depicting the number of events that occurred per second.

4. To view specific events and their properties, click the **Topics** view in the contents panel, select the topic, and then select **View Topic** from the **Actions** menu on the **Activity and Status** pane.

The topic details for the source displays the details of the event, such as the event name, the source type, and the timestamp of the event.

Scenario 2: Viewing Responses

1. Click the **Dashboard** tab.
2. Click the responder controller on the left pane.
3. Click the **Responses** view in the contents panel.

The contents panel displays the number of alerts generated.

4. Select a response, and click **View Responses** from the **Actions** menu on the **Activity and Status** pane.

The details for a particular response, such as the response type sent from the source to an administrator, the time stamp, and the property of the response are displayed.

Scenario 3: Checking Rule Activation

You can check if a rule activates and triggers a response.

1. Click the **Dashboard** tab.
2. To check if a rule is activated, select the event processor where you deployed the rule.
The contents panel displays all the deployed rules, responses, analytics, topics, and watchlists.
3. In the **Rule** view, select the rule to view if the rule is activated for an event.
The contents panel displays the number of activations.
4. To see the alert properties, click the **Responses** view in the contents panel, and then select **View Responses** from the **Actions** menu on the **Activity and Status** pane.

Scenario 4: Purging Events and Alert Details of Objects

You can purge events, activation count, and alert details generated from the system at the component, topology, and object level. All the trace data generated from the source, responder, rule, and event trace is stored in the database. When the volume of trace data generated is high, the system might slow down or might go out of table space. It is recommended that you clear the object metrics from the dashboard every week.

1. Click the **Dashboard** tab.
2. To purge event details from the topology, select **Purge Topology Metrics** from the **Actions** menu in the upper-left pane.
3. To purge details of all objects within a selected application service level, select the application service on the left pane, and then select **Purge Controller Metrics** from the **Actions** menu in the upper-left pane.
4. To purge events for each object, perform the following tasks:
 - a. Select the source controller, event processor, or responder controller.

- b. Select the corresponding views in the contents panel, select the object, and then click **Purge Object Metrics** from the **Actions** menu on the **Activity and Status** pane.

Scenario 5: Using Tracing for Troubleshooting

You can collect trace information for an event to debug an error. Each event includes specific information about the event occurrence, how many events are triggered, and other details that can assist you in diagnosing an event.

1. Click the **Dashboard** tab.
2. To trace a rule and troubleshoot any errors, select the event processor in the left pane.
3. Select the rules in the **Rules** view in the contents panel, and click **Enable Rule Tracing** from the **Actions** menu on the **Activity and Status** pane.
4. To view the activation details after enabling rule tracing, click **View Tracing**.

The view trace report displays the following details:

- The date and time for each incoming event and the topic name.
- Activations display as true when a rule activates.
- Passed simple condition is true for a topic when the condition helps filtering events received on that topic, when considered one at a time.
- When you point the mouse over the rule, you can view the WHEN, WITH, and THEN conditions of the rule. If the activations display as false, the conditions of the rule that the event does not pass is color-coded in red.
- You can point the mouse over **Events** or **Responses** to view the event or response properties.
- You can also view the configured conditions, analytics, and watchlists referenced in the rule.
- The summary displays the total number of evaluations, event sets, activations, and non-activations.

Note: If you enable event tracing and the run-time environment goes down, you need to start the run-time instance, disable event tracing for that rule, and re-enable tracing to continue viewing events in the event trace.

Scenario 6: Creating Events for Troubleshooting

If you encounter an error for an event during any operation, you must enable rule tracing, simulate the event by creating a similar event, and view the event tracing report to troubleshoot the errors.

1. Click the **Dashboard** tab.
2. To manually create an event without using an existing source, click the source controller, and click the **Topics** view.
3. Select **Create Events** from the **Actions** menu on the **Activity and Status** pane.
4. To publish an event for that topic, add a property and value to the topic, and click **Create Event**.

The contents panel displays the number of events for that topic.

Scenario 7: Copying Events for Troubleshooting

If a particular rule evaluation showed errors and you want to debug the evaluation, you must first enable rule tracing, copy the event that resulted in errors, and view the event tracing report to troubleshoot the errors.

1. Click the **Dashboard** tab.
2. To view specific events and their properties, click the source controller on the left pane.
3. Click the **Topics** view in the contents panel, and select the **View Topic** from the **Actions** menu on the **Activity and Status** pane.
4. To copy an event, filter the events either by source type or time stamp, select the event type from the **Topic Details** screen, and click **Copy Event**.

CHAPTER 12

Object Import and Export

This chapter includes the following topics:

- [Object Import and Export Overview, 133](#)
- [Import, 133](#)
- [Export, 135](#)

Object Import and Export Overview

You can use the administrator user interface options to import and export the configured RulePoint objects across projects. Only a user with super user privileges can administer this option.

The objects can include sources, rules, responders, topics, connections, responses, analytics, watchlists, and templates. You can also import users and user roles from a project.

Use the import option to import configured objects into the RulePoint design time. Use the export option to export configured RulePoint objects from a project to an XML file. You can later import the XML file into a RulePoint project when required.

You can export and import objects that are not valid. Examples for objects that are not valid include a responder without a response, a source without a schedule, and templates and rules where the DRQL statements are incorrect.

You cannot import or export secondary objects that are not referenced in the primary objects. If you want to import or export a secondary object, ensure that you reference them in a source, rule, or responder. For example, if you refer a topic and connection in a source, when you export or import the source, the topic and the connection are also exported or imported.

Import

You can import a set of configured primary and secondary objects into RulePoint. You can import only dependent secondary objects.

When you import objects to a project, you can choose to fail, skip, or update collision of similar objects that exist in that project.

You can import any file that exists in the **Available Files** view under **Import**. You can import the files uploaded into the system or files exported from any project.

Uploading a File

You can upload an XML file containing the objects from the native location where you have saved the file into RulePoint. If a similar file exists in the project, the uploaded file is overwritten.

1. On the **Administration** tab, click the **Import** view.
2. From the **Actions** menu in the navigator or in the upper-right pane, click **Upload File**.
3. Under **File Name**, click **Browse** to select the path of the XML file.
4. Click **Upload File**.

A message appears that indicates successful upload of file, as `<file_name.xml>`.

Importing a File

You can import a file containing the configured objects into a project in RulePoint.

Before you import a file, you need to upload the XML file from the native location where you have saved the file into RulePoint. If you want to import objects from a specific project, you must first export the objects from that project before you import it into this project in RulePoint.

1. On the **Administration** tab, click the **Import** view.
2. Select the project where you want to import the objects.
3. In the **Available Files** view in the contents panel, select the uploaded file, and click **Start Import** from the **Actions** menu on the upper-right pane.
4. In the **Import** dialog box, choose one of the following options, based on whether you want to fail, skip, or update collision of objects:
 - Select **Fail** to fail the importing operation if any object that you import already exists in the selected project.
 - Select **Skip** to skip importing an object that is present under the project, but proceeds to import new objects into the project.
 - Select **Update** to overwrite the properties of objects in a project with the properties of the newly imported objects, and import new objects.

Note: The default option is update on collision.

5. Provide a description for the task.
6. Click **Import**.

A message appears that indicates that the import is successful.

7. Click **OK**.

The list and details views displays the summary and details of the imported file, the object types imported, and the status for each object.

Deleting an Imported File and History

You can delete an imported file and the history of an imported file.

1. On the **Administration** tab, click the **Import** view.
2. Perform the following tasks:
 - To delete the history, select the project in the navigator, and in the **Import History** view of the contents panel, select the file for which you want to delete the import history.

- To delete an imported or uploaded file, click the **Available Files** view in the contents panel, and select the file that you want to delete.
3. From the menu, select **Delete**.
A message appears that prompts you if you want to delete.
 4. Click **OK**.

Export

You can export configured objects in the form of an XML file from RulePoint.

You have the option to export all the primary objects and their dependent secondary objects, or export only a selected set of primary and dependent secondary objects. You can also export secondary objects that are not associated to a primary object. When you export the primary objects for use in another project, make sure that you have the required secondary objects.

When you export, you can choose to export the ACLs associated with the exported object. You can also choose to include the system configurations in the XML file. The system configurations in the XML report contain information about the environmental variables.

Exporting Selected Objects

You can export selected objects in a project from the **Actions** menu in the navigator or the **Export Selected** button in the contents pane.

1. On the **Administration** tab, click the **Export** view.
2. From the **Actions** menu in the navigator or in the upper-right pane, select **Export Selected** objects.
3. In the **Export Selected Objects** dialog box, perform the following tasks:
 - a. Enter a file name for the XML file.
 - b. Provide a description for the file.
 - c. To include RulePoint objects that are not valid, select **Include invalid artifacts**.
 - d. To include users associated with the object, select **Include ACLs in Export Set**.
 - e. To include system configurations, select **Include System Configuration in the Report**.
 - f. Click **Next**.
4. Under **Select Objects**, you can perform the following tasks:
 - a. Select all the objects that you want to export, and click the **Include All Dependencies** icon to move the selected objects along with their supporting objects into the **Objects to Export** list on the right panel.
 - b. To export objects of your choice, select that object in the **Objects** pane.
The **Dependent Objects** pane displays the supporting objects that are referenced in the selected primary object. The dependent objects are selected by default. You can choose to select only the dependent objects that you want to export. Click the **Add Selected** icon to move all the selected objects into the **Objects to Export** list on the right panel.
5. View the **Objects to Export** list to verify if the list includes all the objects you want to export, and click **Next**.

Note: You can remove the objects from the **Objects to Export** list. You can choose to select all objects or only those objects you want to remove from the list and select the **Remove Selected** icon to move the item back to the **Objects** list on the left pane.

6. Select the users available in RulePoint that you want to export.

7. Click **Export**.

A confirmation message appears indicating successful export.

8. Click **OK**.

The **Details View** displays a summary of the exported primary and secondary objects, along with the object name, type and status.

Exporting All Objects

You can choose to export all objects from a project. When you export all objects, the users and the roles associated with the users are also exported.

1. On the **Administration** tab, click the **Export** view.

2. From the **Actions** menu on the navigator or the upper-right pane, select **Export All Objects**.

3. Enter a file name.

4. Enter a description for the file.

5. Select the following options according to your requirements:

- Include invalid objects.
- Include ACLs in the export set to include users associated with the objects.
- Include the system configurations in the report.

6. Click **Export**.

A confirmation message appears indicating successful export.

7. Click **OK**.

The file is available for import in the **Available Files** view in the contents panel. The details view displays the summary and details of the exported objects.

Downloading a File

You can download an exported file containing the RulePoint objects.

1. On the **Administration** tab, click the **Export** view.

2. In the contents panel, select the file that you want to download, and click **Download** from the menu.

3. Based on the options supported by the browser, perform the following steps:

- Save the file.
- Open the file and save.
- Open the file by using the listed application, and then save the file.

You can also select to carry out the operation automatically for all future file downloads.

Deleting an Exported File and History

You can delete the exported file and the exported history.

1. On the **Administration** tab, click the **Export** view.
 - To delete the history of an exported file, select the file in the contents panel, and click **Delete** from the menu.
 - To delete an exported file from the disk, select the file in the **Available Files** view on the **Import** tab, and click **Delete** from the menu.

A message appears that prompts you if you want to delete.

2. Click **OK**.

CHAPTER 13

Markers

This chapter includes the following topics:

- [Markers Overview, 138](#)
- [Source Markers, 138](#)
- [Marker Rules, 139](#)
- [System Marker Properties, 139](#)
- [Configuring a Marker, 140](#)
- [Best Practices for Using Markers, 140](#)

Markers Overview

You can add markers to sources to understand the behavior of events. Markers listen to incoming events and provide information about the events. When you send a batch of events into RulePoint, markers help you understand the time it takes to process that batch of events.

A source produces marker events to the topics, if you configure a marker in that source. You can create marker rules to listen to the system markers. When the events match the rule condition, you get an alert that contains information about the event.

Only the pull sources support markers.

Source Markers

If you want more information on sources, you need to add a marker while configuring a source. Provide a marker ID for the source.

You cannot share markers across sources and topics. If two sources produce events to the same topic, each of the topics would have their own start and stop events with their identifier.

The source ID is part of the marker event. If the source controller fails at any point, there is the possibility that you might have duplicate events for the same source. When the system comes up, and the event processor starts processing events again, if there is an existing marker event, it drops the duplicate start and stop markers from the same source based on the ID.

Marker Rules

Marker rules evaluate marker events on a source topic. System marker rules run only on the system marker topics.

You need to create a marker rule to listen to the configured markers and provide a response when the condition satisfies the rule.

Example Marker Rule

If you use an RTAM response, you can use the following rule to test markers:

```
when system_markers then rtamresponse with body="begin time is ${begin_time} ,end time is $
{end_time} , engine name is ${engine_name} ,event count is ${event_count} ,hit ratio is $
{hit_ratio} ,marker name is ${marker_name} , maker_type is $
{marker_type} ,rule_activation_count is ${rule_activation_count} ,rules_processed is $
{rules_processed} ,source list is ${source_list} and total processing time is $
{total_processing_time}
```

System Marker Properties

Marker properties are set in the marker topic (system_markers) when the event processor processes the end event. You cannot change the marker properties.

The following table describes the attributes of system markers that are part of the event:

Name	Description
begin_time	The epoch time when the event processor begins to mark events.
end_time	The epoch time when the event processor stops marking events.
engine_name	Name of the event processor that produces the marker begin and end events.
event_count	Total number of events generated from sources pulled into RulePoint based on a particular schedule. The event count is the total events received between the begin and end time of a marker boundary.
hit_ratio	Ratio of the number of events in the marker boundary that participated in activations and the total number of events. Formula: Hit ratio % = Total number of activations/Total number of events x 100. For example, if there are 100 events, and only 10 events are activated, the hit ratio is 10%. The higher the number of activations, the higher the hit ratio %.
marker_name	The name of the marker that you provide in the source configuration screen.
marker_type	The marker value. It is either the begin event or end event. The begin event is when the source begins processing events, and the end event is when the source stops processing events.
rule_activation_count	Aggregate number of events that are activated for an event within a begin and end event.
rules_processed	A comma-separated list of all rules that processed events received within a marker boundary.

Name	Description
source_list	A comma-separated list of sources that produced the marker events.
total_processing_time	The total processing time, in milliseconds, that the event processor takes to process a batch of events within a marker boundary.

Configuring a Marker

If you add a marker to the source, you must create a rule that listens to the marker topic and send you an alert when the rule matches a configured marker condition.

1. Configure a source to include a marker. Add a unique marker ID for the source in the configuration screen when you create a source.
2. Create a rule for a source, which listens to the topic to which the source produces events.
3. Create a rule for the marker which listens to the system markers.

For information on a sample marker rule, see [“Marker Rules” on page 139](#).

4. Deploy the marker sources, rules that listen to the source topic, the marker rule that listens to the system_markers topic, and responders to the corresponding application services.

When the events match the rule condition, you get an alert that contains information about the event.

Best Practices for Using Markers

Use the following best practices for using markers:

- If there are multiple rules that listen to a source topic, the event processor does not trigger an alert for the marker end event till it evaluates all the rules that listen to the source topic in an event batch. To trigger the batch completion related alert, you need to tune the topic expiry interval to less than the default 3600000 milliseconds.
- Ideally, configure only one rule to listen to a topic so that the event processor triggers a batch completion alert immediately upon the receipt of the marker end event.

CHAPTER 14

Log Management

This chapter includes the following topics:

- [Log Management Overview, 141](#)
- [RulePoint Logs, 141](#)
- [Log File Configurations, 142](#)
- [Viewing the Log Files, 142](#)

Log Management Overview

RulePoint creates log files to record diagnostic information about its operation.

RulePoint uses the Apache log4j utility to log application and debugging messages that you can view to determine the cause of errors or failures. You can configure the log settings in the log4j.xml file located in the RulePoint logs directory. You can configure the file to control where and in what format RulePoint logs messages.

RulePoint Logs

RulePoint generates debug logs for both the design time and run time and sends them to a log file in the `RULEPOINT_HOME` folder.

RulePoint logs provide diagnostic information on the status of the host agent, nodes, grid manager, UM store, and UM lbmrd processes. Each of these are logged to separate files under `RULEPOINT_HOME`. The log files will help you figure out if there are any issues with the RulePoint system. A RulePoint administrator can monitor these files for errors.

The installer also produces log files during and after the installation. You can use these logs to get more information about the tasks that the installer completes and the errors that occur during installation.

Log File Configurations

You can manage the amount of log messages stored in a file and the backup of log messages.

When a log file reaches the maximum size defined in the defaultFileAppender parameter in the log4j.xml configuration, RulePoint archives the existing log information and prevents log files from being overridden.

You need to define the log configuration settings in the log4j.xml file for the design time and the run time at the following location:

- Design time. RULEPOINT_HOME\$\design\webapps\rulepoint\WEB-INF\classes folder.
- Run time. RULEPOINT_HOME\conf folder.

You can configure the following values for log4j.files:

- Fine tune the maximum value of the file size in the defaultFileAppender class to a value greater than the default value of 20 MB.
- Change the maximum file size for the design time to greater than the default value of 10 MB.
- Change the MaxBackupIndex value to more than the default value of 5.
- Change the default log level of INFO for the design-time and run-time logs.

Viewing the Log Files

Error messages are captured and appended to the log file. You can check the logs to determine the cause of an error. The log messages describe all management tasks that a user completes in the run time and design time.

1. Navigate to the RulePoint directory that contains the log files. See the table for locating a particular log file.
2. Open the .log file by using a text editor.

Log File Location

The log files are located in the RulePoint directory.

The following table lists the logs and log locations for RulePoint:

Log Category	File Name	Location
Tomcat	- catalina.2013-07-03.log - host-manager.2013-07-03.log - localhost.2013-07-03.log - localhost_access_log.2013-07-03.txt - manager.2013-07-03.log	RulePoint_6\design\logs
Design time	rulepoint.log	RulePoint_6\logs
RulePoint start and stop	yosemitestartstop.log	RulePoint_6\logs
RulePoint installation	Rulepoint_6_Install.log	RulePoint_6

Log Category	File Name	Location
Grid manager	grid_manager__< topology name >_Grid_Manager.log	RulePoint_6\logs
Node agent	nodeagent.log	RulePoint_6\ nodeagent_<hostname> \logs
UM Store	stored.log	RulePoint_6\ stored.<topology name>_UM-Store\logs
UM lbmrd	lbmrd.<topology name>_UM-Lbmrd.log	RulePoint_6\ lbmrd.<topology name>_UM-Lbmrd

Log Format

All RulePoint log messages follow a particular format.

The following example illustrates the format for a design-time log:

```
<07-08-2013 10:59:25,257 AM +0530> [INFO ] ContextLoader: Root WebApplicationContext: initialization started
```

The following table describes each segment of the message format for the design-time log:

Message Format	Description
Date/Time	The date the message was logged. Format: MM-dd-yyyy hh:mm:ms a Z
[INFO]	The logging level. Valid values include: <ul style="list-style-type: none"> - FATAL - fatal error message - ERROR - error message - WARN - warning - INFO - informational message - DEBUG - debug message
ContextLoader	The service, package, or class name for which the message was logged.
Root WebApplicationContext: initialization started	The message text.

The following example illustrates the log format for a run-time log:

```
[08 Jul 2013 10:59:38,505][SERVER_TopologyConstructionThreadPool_2:][INFO ] [UMLbmrdManager] LBMRD 1 is being launched
```

The following table describes each segment of the message format for the run-time log:

Message Format	Description
Date/Time	The date the message was logged. Format: MM-dd-yyyy hh:mm:ms a Z
[SERVER_TopologyConstructionThreadPool_2:]	The executing thread name.

Message Format	Description
[INFO]	<p>The logging level. Valid values include:</p> <ul style="list-style-type: none"> - FATAL - fatal error message - ERROR - error message - WARN - warning - INFO - informational message - DEBUG - debug message
[UMLbmrManager]	The service, package, or class name for which the message was logged.
LBMRD 1 is being launched	The message text.

CHAPTER 15

Licenses

This chapter includes the following topics:

- [Licenses Overview, 145](#)
- [License Validation, 145](#)

Licenses Overview

The license file contains the license key, which is in encrypted form. The grid manager decrypts the license key and enables the purchased options. The license for each customer would vary based on the number of years that the customer plans to use the product.

License Validation

When you start RulePoint, the grid manager checks for the validity of the license present in the configuration directory, before it proceeds with any operation.

If the license check succeeds, RulePoint starts up. Else, the grid manager brings down the system, and sends an email notification to the administrator.

The grid manager also schedules a periodic check for validating the license, every 24 hours. If the license reaches its expiration time, the grid manager sends an email to the administrator.

APPENDIX A

Error Codes

This appendix includes the following topics:

- [Design-Time Error Codes, 146](#)
- [Interaction Error Codes, 149](#)
- [Import Error Codes, 150](#)
- [Export Error Codes, 151](#)
- [Security Error Codes, 151](#)
- [ACL Error Codes , 152](#)
- [Run-Time Error Codes, 153](#)

Design-Time Error Codes

The following table provides the error codes, description, and resolution for the design-time errors:

Error Code	Description
YDB-1001	No deployment plan exists for Project with ID {0}.
YDB-1003	Define a topology to invoke RunOnce.
YDB-1005	Delete failed for {0} with id {1}.
YDB-1006	The object or action specified by the URL does not exist on the RulePoint server.
YDB-1008	The {0} with the id {1} could not be found.
YDB-1009	Object testing not supported.
YDB-1011	Conversion from {0} to {1} failed.
YDB-1012	{0} cannot be null.
YDB-1013	Cannot find any operations for web services with name {0} in the WSDL: {1}.
YDB-1014	Unable to parse the url: {0}.
YDB-1015	An error occurred while compiling DRQL: {0}.

Error Code	Description
YDB-1016	No value is provided for the required property [{0}], at type: {1}.
YDB-1018	Creation of {0} failed.
YDB-1019	The template configuration is not valid.
YDB-1020	Failed to test the object.
YDB-1021	The rule configuration is not valid.
YDB-1023	An internal server error has occurred.
YDB-1024	The value of the property {0} of type {1} in {2} is not valid.
YDB-1025	The watchlist type [{0}] is not valid.
YDB-1026	The RunOnce task failed for the source [{0}] with an exception - {1}.
YDB-1027	Data Integrity Failure.
YDB-1028	No bean information found for {0}.
YDB-1029	Cannot find service with the name [{0}] in the WSDL. {1}
YDB-1030	The required primary objects are missing in the redeploy request.
YDB-1031	The wizard configuration is not valid.
YDB-1032	Operation for the object [object name - {0} type - {1} state - {2}] failed because its deployment is still underway.
YDB-1034	The operation [{0}] is not allowed on the object {2} currently in state {2}.
YDB-1035	Empty Analytic Map
YDB-1036	The operation [{0}] is not allowed on the predefined object [{1}].
YDB-1037	Object {0} is not valid.
YDB-1039	The request parameter contains a value that is not valid.
YDB-1040	Validation Error! Property Path: {0}, Root Bean: {1}.
YDB-1041	The value of the parameter missing in the template {0}: {1}.
YDB-1044	Value {0} for parameter {1} is not valid: {2}.
YDB-1045	The Topic {0} is not allowed for Source {1}.
YDB-1046	No services found for the WSDL.
YDB-1047	Failed to store the schedule.
YDB-1048	The given value of property : {0}, at type: {1} does not match the expected pattern {2}.

Error Code	Description
YDB-1049	The request parameter {0} is missing.
YDB-1050	Update failed for object {0} with ID {1}.
YDB-1051	Cannot create a schedule on a non-schedulable source [{0}].
YDB-1052	Conversion from {0} to {1} is complete.
YDB-1053	Provide a topic for the source.
YDB-1054	Cannot find the project with ID {0}.
YDB-1055	{0} {1} has {2} dependencies.
YDB-1055	{0} {1} has {2} dependencies: {3}.
YDB-1056	The profiler output could not be stored.
YDB-1057	Cannot find service type [{0}].
YDB-1058	Attempting to create duplicate key for {0} - {1}. Another {0} with same identifier present in the system.
YDB-1059	Deployment of Object(s) failed.
YDB-1060	{0} type beanInfo not found for {1}.
YDB-1061	Cannot find schedules for [{0}].
YDB-1062	You can deploy only when the referrer to a supporting object is a primary object.
YDB-1063	The group names [{0}] are not valid/could not be validated for deployment of {1} {2}.
YDB-1064	Action [{0}] is not allowed on object [{1}] of type [{2}] in [{3}] state.
YDB-1065	Deployment attempted on template rules for template {0} that has no deployment policy.
YDB-1066	No objects are found in the deployment package.
YDB-1067	{3} failed for objects that are not valid. Object[name - {0} type - {1} state - {2}] is not in valid state.
YDB-1068	Required dependent class is missing. Verify that the required jar is in the classpath.
YDB-1069	Testing of object {0} is not supported.
YDB-1070	No object was found in request to be copied.
YDB-1071	Action is not allowed - {0} - when {1}.
YDB-1072	Enter a start time with a valid value.
YDB-1073	You defined an invalid date for the schedule. Define an end time after the start time.
YDB-1074	The template has following dependencies: {0}.

Error Code	Description
YDB-1075	Project contains another [{0}] with the same name. Use a different name.
YDB-1076	Cannot find the watchlist [{0}].
YDB-1077	You can create only one dynamic schedule for a source.
YDB-1078	The reassign operation failed for objects in the project [{0}]. Define a deployment plan.
YDB-1079	The reassign operation is not applicable for template rules.
YDB-1080	The {0} operation failed as Controller Group Names are not provided for primary object {1} in the request.
YDB-1081	SQL query execution has timed out.
YDB-1082	The redeploy of supporting object operation failed. Cannot find related primary objects in the NEEDS_DEPLOYMENT state.
YDB-1083	Value of incorrect size provided for property [{0}], at type [{1}]. Size must be between [{2}] and [{3}].
YDB-1084	Only special characters such as space, dot, and hyphen are allowed. Names must start with a letter.
YDB-1085	Special characters are not allowed in the name. Names must start with a letter.
YDB-1086	No special characters except space are allowed in name. Names must start with a letter.
YDB-1087	No special characters except dot and hyphen are allowed. Names must start with a letter.
YDB-1088	Value [{0}] provided for the object name [{1}] is a RulePoint keyword. Use a different value for the object name.
YDB-1089	The value of parameter [{1}] in the template rule [{0}] is missing.
YDB-1090	Template DRQL has changed. Perform an upgrade.
YDB-1091	RunOnce cannot be invoked on a non schedulable source.

Interaction Error Codes

The following table provides the error codes, description, and resolution for errors generated during design-time and run-time interaction:

Error Code	Description
YDR-3001	There was an error while trying to retrieve Topology details - {0}.
YDB-3002	No event processors are found in the topology with ID [{0}] for the default deployment policy creation for template [{1}].

Error Code	Description
YDB-3003	No event processors found in topology with ID :- {0}
YDB-3004	No source or responder controllers found in the topology with ID {0}
YDB-3005	A runtime error occurred.[Code - {0} Message - {1}]
YDB-3006	A runtime error occurred.
YDB-3007	The Grid Manager is not reachable.
YDR-3008	Search Operation Failed.
YDR-3009	An error occurred in Log viewer Module

Import Error Codes

The following table provides the error codes, description, and resolution for the import errors:

Error Code	Description
YDM-4000	Import failed. Conversion failed for {0} : {1} Property : {2}.
YDM-4001	Object name [{0}] of type [{1}] already exists.
YDM-4002	Import failed! Validation failed for {0} : {1} Property : {2}
YDM-4003	Incorrect ACL Entries found in the XML for [{0}:{1}].
YDM-4004	Cannot find {0}:{1}; property:{2} in the XML.
YDM-4005	Cannot find {0}:{1} property in the XML.
YDM-4006	While importing, {0}:{1} property:{2} cannot be null.
YDM-4007	While importing [{0} : {1}], it must refer to at least one topic.
YDM-4008	While importing a rule [{0} : {1}], the rule must refer to at least one response.
YDM-4009	The object properties [{0}]:{1} {2} : {3}] for importing is not found in the XML.
YDM-4010	Import failed.
YDM-4011	Failed to import objects because the service could not be validated.
YDM-4012	Import failed.
YDM-4013	Error uploading the file.
YDM-4014	Error reading the XML file report data!

Error Code	Description
YDM-4015	Error reading available file for import metadata. {0}.
YDM-4016	Import conversion failed!
YDM-4017	Conversion of {0} failed!. {1} not found.
YDM-4018	Object [{0}] should have at least one {1}.
YDM-4019	Import failed!
YDM-4020	Import of user role [{0}] : {1} information failed.
YDM-4021	Import of user/role failed!
YDM-4022	Import failed because the XML file is not compatible with the current version of RulePoint. The file was generated using RulePoint version {0}
YDM-4023	Template params not found in the XML for {0}:{1}.

Export Error Codes

The following table provides the error codes, description, and resolution for the export errors:

Error Code	Description
YDM-4201	Export failed!
YDM-4202	Exporting service configuration failed for {0} : {1}.
YDM-4203	Export failed because file [{0}] has invalid objects.
YDM-4204	There are no objects to export for {0}.
YDM-4205	Select a valid option for exporting object {0}.

Security Error Codes

The following table provides the error codes, description, and resolution for the security error codes:

Error Code	Description
YDS-1501	User {0} does not have the necessary authorization for action {1} on object type {2} item {3}.
YDS-1502	User {0} does not have the necessary authorization for the requested action {1}.

Error Code	Description
YDS-1503	Passwords must be between 9 to 16 characters, contain only letters and numbers, and contain at least 1 letter and 1 number.
YDS-1504	If you are currently logged in, you cannot delete your user credentials.
YDS-1505	Provide a remoteUniqueName for the remote user.
YDS-1506	The remoteUniqueName value {0} provided is not a valid user DN.
YDS-1507	LDAP authentication has not yet been configured or has been disabled for this setup. This action is only available after LDAP is configured and enabled.

ACL Error Codes

The following table provides the error codes, description, and resolution for ACL errors:

Error Code	Description
YDB-1801	The ACL Owner object is not found.
YDB-1802	An error occurred in the ACL operation. - {0}.
YDB-1803	Cannot find the user name mentioned in the ACL entry.
YDB-1804	ACL data not found in database.
YDB-1805	ACL data to be updated is null.
YDB-1806	Duplicate ACL Entry given [User-{0} , Granting={1} , Permission={2}].
YDB-1807	You can not edit self entry in the ACL.
YDS-1808	The action {1} on Role {0} has failed because this Role and/or remote users of Ldap Groups mapped to this Role has ACL/ACL Entries. Please specify appropriate targets for transfer of these ACL Entries and try again.
YDS-1809	The User {0} , to be deleted has ACLs/ACL entries. Please specify the username/rolename to transfer these ACLs/ACL entries to and try again.
YDS-1810	Invalid user/role name {0} provided for ACL purge on Role {1}.
YDS-1811	Invalid user/role name {0} provided for role ACL transfer on Role {1}. Either no user/role by name {0} exists or it is same as the source role.
YDS-1812	Replacement Sid {0} is same as original Sid {1}.
YDS-1813	Invalid user/role name {0} provided for ACL purge on user {1}.

Run-Time Error Codes

The following table provides the error codes, description, and resolution for the run-time errors.

Error Code	Description
YRM-5000	Error occurred while processing the request.
YRM-5001	Error occurred while accessing Database. Cause {0}.
YRM-5003	Database url is missing in the configuration file. Please add the url.
YRM-5004	Database driver class is missing. Please add the driver class.
YRM-5005	Database username is missing. Please add the username.
YRM-5006	Database password is missing. Please add the password.
YRM-5007	Error occurred while loading driver class of the Database.
YRM-5008	Database is not accessible.
YRM-5120	Majority({0}) of the UM stores have failed. Runtime will be shutdown.
YRM-5125	UM topic resolution daemon - Lbmrd - has failed. Runtime will be shutdown.
YRM-5140	Remote Host {0} threw an error.
YRT-8141	Remote Host {0} threw a transport error.
YRM-5150	Remote Node {0} threw an error.
YRM-5151	Remote Node {0} threw an error while starting deployment.
YRM-5152	Remote Node {0} threw an error while ending deployment.
YRT-8155	Remote Node {0} threw a transport error.
YRM-5160	Remote Grid Manager {0} threw an error.
YRT-8161	Remote Grid Manager {0} threw a transport error.
YRM-5201	Grid Manager cannot process request since it is not running in primary mode or there has been a fatal error.
YRM-5202	Could not get IP address for the Grid Manager process.
YRM-5203	Resource file {0} is not found in the conf folder.
YRM-5205	Service type {0} got from Topology configuration is unknown.
YRM-5206	High availability mode {0} got from Topology configuration is unknown.
YRM-5207	Grid Manager failed to start since port {0} is already bound.
YRM-5210	Service cannot be set to primary since node is not running on the configured host.

Error Code	Description
YRM-5211	Topology is being decommissioned. Cannot process request.
YRM-5212	Topology is being shut down. Cannot process request.
YRM-5213	Topology has not started. Cannot process request.
YRM-5214	Deployment is in progress. Please wait for it to finish and then retry.
YRM-5215	Could not failover to available services configured in the high availability group {0}. Attempting to make the restarted service as primary.
YRM-5216	License has expired. Please contact your system administrator.
YRM-5217	License file not found. Please contact your system administrator.
YRM-5218	Cannot start Topology from this Host. Topology can be started only from a Host that has the Grid Manager assigned to it.
YRM-5219	Cannot start Topology from this Host. Please ensure that this Host is part of the Topology.
YRM-5221	An active Grid Manager is not running on this Host.
YRM-5222	Grid Manager cannot process request since it is not running in active mode.
YRM-5223	Error occurred while getting an instance of Configuration service. Cause {0}.
YRM-5224	Configuration service threw an error. Cause {0}.
YRM-5225	An error occurred while stopping {0}. Cause {1}.
YRM-5226	An error occurred while starting {0}. Cause {1}.
YRM-5227	An error occurred while initializing {0}. Cause {1}.
YRM-5302	Could not determine if the Host is local.
YRM-5303	Could not launch the node process. Cause {0} .
YRM-5304	Execution Unit doesn't exist in the system. Cannot remove nonexistent EU.
YRM-5400	Deployment failed for the following object {0}. Cause : Dependent object {1} is not found.
YRM-5401	Deployment failed for the following object {0}. Cause : {1}.
YRM-5403	The specified Deployment Group is not present in the Topology. Please check the group name {0}.
YRM-5405	Deployment Group is not found for object {0}, {1} of type {2} in the Topology.
YRM-5406	Deployment package {0} already exists in the Grid Manager. Please use a different Id.
YRM-5407	Deployment failed for object {0} because it already exists.
YRM-5408	Update/Un-deploy failed for object {0} because it does not exist.

Error Code	Description
YRM-5409	Transactional attribute {0} is not supported.
YRM-5410	Deployment failed since no service from group {0} is running.
YRM-5411	Deployment failed due to {0}.
YRM-5412	Start failed for object {0} because it does not exist.
YRM-5413	Stop failed for object {0} because it does not exist.
YRM-5414	Only primary objects can be started.
YRM-5415	Only primary objects can be stopped.
YRM-5416	Object {0} is already in started state.
YRM-5417	Object {0} is already in stopped state.
YRM-5418	Deployment action {0} for object {1} is not supported.
YRM-5419	RunOnce is supported only on source objects.
YRM-5420	Object {0} is not in a started state.
YRM-5421	Object {0} is not running in the selected group.
YRM-5422	Object should be of type Topic to generate event.
YRM-5423	An error occurred while reverting previous unfinished deployments.
YRM-5424	An error occurred while deploying the package {0}.
YRM-5425	An error occurred while checking the dependencies. Cause {0}
YRE-7500	Event Processor is not initialized.
YRE-7501	Event Processor is already initialized.
YRE-7502	Invalid Event Processor configuration {0}.
YRE-7503	Event Processor could not be stopped in time.
YRE-7504	Incorrect object configuration.
YRE-7520	Rule is already running. Previous rule-execution may not have been stopped. Please retry after sometime.
YRE-7521	Rule is not running.
YRE-7522	Rule is not deployed.
YRE-7523	Rule is already deployed.
YRE-7524	Rule conditions cannot be instantiated.

Error Code	Description
YRE-7526	Failed to start Rule with ID {0}.
YRE-7560	Invalid {0} configuration {1}.
YRE-7550	Analytic failed, Reason : {1}.
YRE-7551	Analytic failed, Reason : Exception in number conversion.
YRE-7552	Analytic failed, Reason : Null argument passed.
YRE-7553	Analytic failed, Reason : Invalid arguments/parameters - {1}.
YRE-7560	Invalid {0} configuration {1}.
YRE-7561	Object {0} with ID {1} already exists.
YRE-7562	Missing dependent object {0}.
YRE-7563	Cannot remove {0}. There are dependent object(s) {1}.
YRE-7564	Cannot instantiate class {0}.
YRE-7565	Cannot initialize {0}.
YRE-7566	Cannot close connection in {0}.
YRE-7567	Cannot establish connection in {0}.
YRE-7580	Secondary Event Processor: Book keeping service got duplicate activation id.
YRE-7581	Secondary Event Processor: Book keeping service got event out of order.
YRE-7582	Secondary Event Processor: Book keeping service got processed event id out of order.
YRT-8600	UM LBM initialization error.
YRT-8601	UM source Topic is not found {0}.
YRT-8602	UM receiver Topic is not found {0}.
YRM-5700	Error while starting service.
YRM-5701	Error while stopping service.
YRM-5702	Error while changing high availability mode of a service.
YRM-5703	Error in execute call.
YRM-5705	Error in creating/deleting response UM topic.
YRM-5706	Invalid ID prefix {0}.
YRM-5710	Invalid object configuration.

Error Code	Description
YRM-5720	Could not create Source Controller.
YRM-5721	Could not start Source Controller.
YRM-5722	Could not stop Source Controller.
YRM-5723	Execute call on Source Controller failed.
YRM-5724	High availability mode of Source Controller cannot be changed from primary to secondary.
YRM-5730	Could not create Responder Controller.
YRM-5731	Could not start Responder Controller.
YRM-5732	Could not stop Responder Controller.
YRM-5733	Execute call on Responder Controller failed.
YRM-5734	High availability mode of Responder Controller cannot be changed from primary to secondary.
YRM-5740	Could not create Event Processor.
YRM-5741	Could not start Event Processor.
YRM-5742	Could not stop Event Processor.
YRM-5743	Execute call on Event Processor failed.
YRM-5744	High availability mode of Event Processor cannot be changed from primary to secondary.
YRM-5745	No matching begin transaction.
YRM-5746	Duplicate begin transaction.
YRM-5760	Service Id should be greater than zero. Id: {0}.
YRM-5761	A Service with id {0} already exists.
YRM-5762	No Service exists with id {0}.
YRM-5763	Unable to get sessionId from metadata.
YRM-5764	SessionId must be greater than zero.
YRM-5765	Service does not support deployment of objects.
YRM-5766	Transport configuration not found in configuration map.
YRM-5767	Invalid configuration value for parameter {0}.
YRM-5768	Invalid Node id.
YRM-5769	Service type {0} is invalid.

Error Code	Description
YRM-5770	Operation RunOnce is not supported on {0}.
YRM-5771	Operation Create Event not supported on {0}.
YRM-5772	Search Operation Failed, Reason : {0}.
YRM-5773	Purge Index Operation Failed, Reason : {0}.
YRT-5780	Could not create transport configuration file.
YRM-5901	An error occurred in Activity Manager while getting entity of type {0}.
YRM-5902	An error occurred in Activity Manager while getting entity of type {0} and identifier {1}.
YRM-5903	An error occurred in Activity Manager while saving a batch of entities of type type {0}.
YRM-5904	Could not get an instance of Activity Manager. Cause {0}.
YRM-5800	Process could not bind to Host and port.
YRM-5801	Valid Host name must be specified.
YRM-5802	Valid port must be specified.
YRM-5803	Invalid metadata passed.
YRM-5804	Unknown error while creating a process.
YRM-5806	UM store configuration is missing.
YRM-5807	Failed to save UM store configuration.
YRM-5808	Valid process identifier needs to be specified.
YRM-5809	Failed to save process state.
YRM-5810	Failed to delete process state.
YRM-5811	Could not find process.
YRM-5812	Duplicate process identifier.
YRM-5813	Failed to stop {0} process {1}.
YRM-5814	Failed to create file/directory {0}.
YRM-5815	Pre-probe check of process failed.
YRM-5816	Valid name needs should be provided.
YRM-5817	A process is already running on port {0}. Cannot start Grid Manager {1}.
YRS-6000	Invalid {0} configuration, {1}.

Error Code	Description
YRS-6001	Object {0} with ID {1} already exists.
YRS-6002	Addition of Object {0} with ID {1} failed with {2}.
YRS-6003	Object {0} with ID {1} reason {2}.
YRS-6004	Invalid Object {0} configuration with ID {1} reason {2}.
YRS-6005	Object {0} configuration has failed, reason {1}.
YRS-6006	Object {0} RunOnce has failed, reason {1}.
YRS-6007	Object {0} connection {1}, reason {1}.
YRS-6008	Object adapter class {0} failed, reason {1}.
YRS-6009	Cannot initialize {0}.
YRS-6010	Cannot close connection {0}.
YRS-6011	Cannot establish connection {0}.
YRS-6012	Object {0} is not enabled, reason {1}.
YRS-6013	Object {0} is not available, reason {1}.
YRS-6500	Framework method {0} failed, reason {1}.
YRS-6510	Scheduler framework, method {0}, reason {1}.
YRS-6300	Object ID {0} with name {1} of type {2} failed at {3}, reason {4}.

APPENDIX B

Glossary

activation

An invocation of a response when an event matches a rule condition.

activity manager

An application service that records the activity of the topology components. The activity manager displays the data recorded on the dashboard of the RulePoint user interface.

analytic

An object that implements a data processing function when it is referenced in a rule.

application service

A service that runs on a node in the run-time environment. Application services include the source controller, event processor, responder controller, and activity manager.

connection

An object that contains information that the RulePoint services use to connect to the target database.

deployment

A process that adds configured objects to an application service that processes the objects. Sources and supporting objects are deployed in the source controller, responders and supporting objects are deployed into the responder controller, while rules and supporting objects are deployed into the event processor.

event

The data that is pulled or pushed into RulePoint from a source.

event processor

An application service that manages the lifecycle of topics, rules, analytics, watchlists, and responses that are deployed to it.

event tracing

A process that collects troubleshooting information related to an event, such as the event occurrence and the number of events that are triggered.

grid manager

A system service that manages nodes, hosts, application services, and system services in a topology.

host agent

A component of RulePoint that runs on each host in a topology and manages communication between the grid manager and the nodes on a host.

marker

An event type that provides information to the event processor regarding the start and end of events in an event execution cycle.

marshaller

An entity that enriches events and activations by adding or removing properties. A marshaller is attached to sources to enrich the generated events before they are further processed by the event processor. A marshaller is attached to responders to enrich the activations before they are processed further by the responder.

node

A JVM process that manages the application services.

responder

An object that invokes a response to an external system such as an email service that notifies specific users of events.

responder controller

An application service that manages the lifecycle of responders, responses, and connections.

response

A configurable action that is invoked by specific conditions set by a rule.

source

An object that has a configurable topic and can be scheduled to run at specific times, such as a news reader that extracts events from an RSS or Atom news feed.

source controller

An application service in the topology that manages the lifecycle of the source, connections, and topics.

system services

A service that runs on the host machine in the run-time environment. System services include the grid manager, UM store, and UM lbmrd.

template

A DRQL rule statement with parameterized variables and instructional text to define the parameters.

topic

An object that logically groups events produced by a source.

topology

A group of application services, system services, nodes, and hosts grouped together based on administrative ownership. A topology constitutes the run-time environment in RulePoint and is the fundamental administrative unit in RulePoint.

UM lbmrd

A system service that resolves addresses for data exchange across the application services.

UM store

A system service that enables data exchange across the application services. The application services use the UM store to persist events to the configured file system.

watchlist

An object that stores multiple values as a single object with a unique name. Rules with a watchlist reference use the data stored in a watchlist to evaluate an event property.

INDEX

A

- accounts
 - managing [20](#)
 - super user privileges [20](#)
- ACL object
 - ACL entry [35](#)
- activity manager
 - statistics [16](#)
 - viewing [32](#)
- add
 - grid manager [81](#)
 - host [54](#)
 - node [57](#)
 - UM lbmrd [33, 83](#)
 - UM store [33, 85](#)
- administration
 - export [28](#)
 - import [28](#)
 - topology [28](#)
 - user interface [27](#)
 - user management [28](#)
- application services
 - event processor [15](#)
 - high availability [18](#)
 - overview [60](#)
 - partitioning [18](#)
 - responder controller [15](#)
 - scaling [18](#)
 - source controller [15](#)
- authentication
 - LDAP [19](#)
 - users [19](#)

B

- best practices
 - deployment [105](#)
 - high availability [96](#)
 - markers [140](#)
 - user management [45](#)

C

- configure
 - event processor [69](#)
 - grid manager [79](#)
 - high availability [89](#)
 - markers [140](#)
 - node [56](#)
 - responder controller [65](#)
 - UM lbmrd [83](#)
 - UM store [85](#)

- configuring
 - source controller [61](#)
- create
 - role [44](#)
- create events
 - dashboard [131](#)

D

- dashboard
 - creating events [131](#)
 - enabling tracing [131](#)
 - event processor configurations [124](#)
 - functions [117](#)
 - overview [19, 117](#)
 - purging events [130](#)
 - responder controller configurations [126](#)
 - rule activation [130](#)
 - source controller configurations [122](#)
 - task types for event processor [127](#)
 - task types for source controller [127](#)
 - task types for topology [129](#)
 - troubleshooting [129](#)
 - use case [129](#)
 - viewing events [129](#)
 - viewing responses [130](#)
 - viewing topics [129](#)
- default topology
 - process flow [18](#)
- delete
 - export history [137](#)
 - exported file [137](#)
 - imported file [134](#)
 - template [110](#)
 - users [43](#)
- deploy
 - all objects [109](#)
 - best practices [105](#)
 - deployment options [102](#)
 - grid manager [16](#)
 - multiple responders [108](#)
 - multiple rules [108](#)
 - multiple sources [107](#)
 - overview [106](#)
 - prerequisites [106](#)
 - primary objects [98](#)
 - responder [108](#)
 - responders [17](#)
 - rule [107](#)
 - rules [17](#)
 - scalability [98](#)
 - source [107](#)
 - sources [17](#)
 - supporting objects [98](#)
 - tasks [100](#)

- deploy (*continued*)
 - workflow [102](#)
- deployment
 - overview [98](#)
 - scenarios [103](#)
- deployment policy
 - template [109](#)
- design time
 - creating [14](#)
 - creating objects [14](#)
 - deploying [14](#)
- download
 - file [136](#)

E

- edit
 - event processor [71](#)
 - grid manager [81](#)
 - host [54](#)
 - node [57](#)
 - responder controller [67](#)
 - source controller [63](#)
 - UM lbmrd [84](#)
 - UM store [86](#)
 - users [43](#)
- event processor
 - configuring [69](#)
 - create in high availability mode [71](#)
 - create in standalone mode [70](#)
 - editing [71](#)
 - failover [94](#)
 - properties [69](#)
 - removing [72](#)
 - rule [15](#)
 - tracing properties [71](#)
 - viewing [72](#)
- export
 - all objects [136](#)
 - file [135](#)
 - overview [133](#)
 - selected objects [135](#)

F

- failover
 - event processor [94](#)
 - grid manager [91](#)
 - primary [19](#)
 - responder controller [92, 93](#)
 - scenarios and action [90](#)
 - source controller [92, 93](#)
- file
 - delete [137](#)
 - deleting [134](#)
 - downloading [136](#)
 - exporting [135](#)
 - import [133](#)
 - importing [134](#)

G

- grid manager
 - adding [81](#)
 - bookkeeper topic [95](#)

- grid manager (*continued*)
 - configuration properties [79](#)
 - configuring [79](#)
 - deploying [16](#)
 - editing [81](#)
 - failover [90, 91](#)
 - functions [78](#)
 - high availability [92, 95](#)
 - ordered topic [95](#)
 - removing [82](#)
 - validating [145](#)
 - viewing [82](#)

H

- high availability
 - application services [19](#)
 - best practices [96](#)
 - configuring [89](#)
 - event processor [95](#)
 - failover [89](#)
 - grid manager [92](#)
 - overview [88](#)
 - recovery [88](#)
 - resilience [88](#)
 - responder controller [93](#)
 - restart and failover [88](#)
 - run time [14](#)
 - source controller [93](#)
 - system services [19](#)
- horizontal scaling
 - run time [18](#)
- host
 - adding [54](#)
 - editing [54](#)
 - host properties [54, 55](#)
 - physical machine [15](#)
 - viewing [55](#)
- host agent
 - instance [22](#)

I

- import
 - file [134](#)
 - overview [133](#)
- importing
 - file [133](#)
- instance
 - design time [23](#)

L

- LDAP
 - authentication [19](#)
- licenses
 - overview [145](#)
 - validating [145](#)
- lifecycle
 - rule [15](#)
- local user
 - password [25](#)
- log
 - LDAP user [25](#)
 - local user [25](#)

- log files
 - configurations [142](#)
 - maximum backup index [142](#)
 - maximum file size [142](#)
 - viewing [142](#)
- logs
 - file location [142](#)
 - log format [143](#)
 - overview [141](#)

M

- manage
 - accounts [20](#)
 - privileges [34](#)
 - roles [34](#), [42](#)
 - users [28](#), [29](#), [34](#), [42](#)
- managing
 - topology [46](#)
- marker
 - properties [139](#)
- markers
 - best practices [140](#)
 - configuring [140](#)
 - examples [139](#)
 - overview [138](#)
 - rules [139](#)
 - source [138](#)

N

- node
 - adding [57](#)
 - configuring [56](#)
 - editing [57](#)
 - functions [56](#)
 - JVM option [57](#)
 - node properties [57](#), [58](#)
 - remove [56](#), [58](#)
 - viewing [58](#)

O

- object
 - deployed [100](#)
 - draft [100](#)
 - needs_deployment [100](#)
- objects
 - deploying [17](#), [109](#)
 - exporting [135](#), [136](#)
 - privileges [35](#)
 - processing [14](#)
 - reassigning [115](#), [116](#)
 - redeploying [110](#)
 - state [100](#)
- overview
 - logs [141](#)

P

- password
 - changing [25](#)
- permissions
 - ACL rules [37](#)

- permissions (*continued*)
 - ACLs [36](#)
 - admin [36](#)
 - deny [36](#)
 - execute [36](#)
 - grant permission [36](#)
 - read [36](#)
 - requirements [37](#)
 - write [36](#)
- permissions users [34](#)
- privileges
 - manage [34](#)
 - objects [35](#)
- process flow
 - default topology [18](#)
- properties
 - event processor [69](#)
 - responder controller [65](#)

R

- real-time
 - processing [13](#)
- reassign
 - multiple responders [116](#)
 - multiple sources [115](#)
 - objects [115](#), [116](#)
 - responder [115](#)
 - source [115](#)
- redeploy
 - all objects [113](#)
 - multiple responders [112](#)
 - multiple rules [112](#)
 - multiple sources [111](#)
 - objects [110](#)
 - responder [112](#)
 - rule [111](#)
 - source [111](#)
- remote users
 - creating [42](#)
- remove
 - event processor [72](#)
 - grid manager [82](#)
 - responder controller [69](#)
- removing
 - node [56](#), [58](#)
 - UM store [87](#)
- responder
 - deploying [108](#)
 - reassigning [115](#), [116](#)
 - redeploying [112](#)
 - undeploying [114](#)
- responder controller
 - application service [16](#)
 - configuring [65](#)
 - create in high availability mode [67](#)
 - create in standalone mode [66](#)
 - editing [67](#)
 - failover [92](#)
 - properties [65](#)
 - removing [69](#)
 - response [16](#)
 - viewing [68](#)
- responders
 - undeploying [114](#)
- roles
 - creating [44](#)

roles (*continued*)
managing [34, 42](#)

rule
deploying [107](#)
redeploy [112](#)
redeploying [111](#)
undeploying [113](#)

RulePoint
overview [27](#)

rules
markers [139](#)
undeploying [114](#)

run time
application services [14](#)
execution infrastructure [14](#)
instance [24](#)
partitioning [19](#)
system services [14](#)

S

scalability
run time [14](#)

scenarios
topology [48](#)

service controller
responder controller [60](#)
source controller [60](#)

services
scaling [13](#)

source
deploying [107](#)
marker properties [139](#)
markers [138](#)
reassign [115](#)
reassigning [115](#)
redeploying [111](#)
undeploying [113](#)

source controller
application services [15](#)
configure [61](#)
create in high-availability mode [62](#)
create in standalone mode [62](#)
editing [63](#)
failover [92](#)
lifecycle [15](#)
properties [61](#)
removing [64](#)
viewing [63, 64](#)

sources
undeploy [113](#)

super user privileges
administrator [35](#)

system services
overview [78](#)
physical processes [16](#)

T

task types
task types for responder controller [128](#)

template
create deployment policy [109, 110](#)
deleting [110](#)

topology
application services [16](#)

topology (*continued*)
configuring [15](#)
considerations [48](#)
default [16](#)
designs [47](#)
host [15, 46](#)
host agent [15](#)
node [15, 46](#)
planning [46](#)
properties [50](#)
requirements [46](#)
scenarios [48](#)
source controllers [46](#)
system services [16](#)
um lbmrd [46](#)
um store [46](#)
viewing [30](#)
tracing dashboard [131](#)
tracing properties
event processor [71](#)
troubleshooting
copy events [132](#)
creating events [131](#)
using tracing [131](#)

U

UM lbmrd
adding [33, 83](#)
address resolution service [16](#)
configuring [83](#)
editing [84](#)
unicast topic resolution [83](#)
viewing [84](#)

UM store
adding [33, 85](#)
configuring [85](#)
editing [86](#)
overview [85](#)
persisting events [16](#)
remove [87](#)
ultra messaging persistence [16](#)
viewing [86, 87](#)

undeploy
all objects [114](#)
multiple responders [114](#)
multiple rules [114](#)
multiple sources [113](#)
objects [113](#)
responder [114](#)
rule [113](#)
running objects [104](#)
source [113](#)

use case
dashboard task types [129](#)

user management
best practices [45](#)

users
creating remote users [42](#)
deleting [43](#)
editing [43](#)
managing [28, 29, 34, 42](#)
permissions [34](#)
privileges [35](#)
role admin [34](#)
role user [34](#)

V

vertical scaling
 run time [18](#)

view

 activity manager [32](#)
 event processor [72](#)
 grid manager [82](#)
 host [55](#)
 log files [142](#)
 node [58](#)
 responder controller [68](#)

view (*continued*)

 source controller [63](#)
 source controllers [64](#)
 topology [30](#)
 UM lbmrd [84](#)
 UM store [86, 87](#)
view events
 dashboard [129](#)
view responses
 dashboard [130](#)
view topics
 dashboard [129](#)