



Informatica™

Informatica® RulePoint
6.1.2

Command Reference

Informatica RulePoint Command Reference

6.1.2

April 2015

© Copyright Informatica LLC 1998, 2018

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 © 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. Copyright © Scalable Systems, Inc. All rights reserved. Copyright © jqWidgets. All rights reserved. Copyright © Tableau Software, Inc. All rights reserved. Copyright © MaxMind, Inc. All Rights Reserved. Copyright © TMate Software s.r.o. 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/hsqLicense.html>, <http://>

<http://unit.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/license.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.sl4j.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/license.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.sl4j.org/license.html>; <http://www.iodbc.org/dataspace/iodbc/wiki/IODBC/License>; <http://www.keplerproject.org/md5/license.html>; <http://www.toedter.com/en/jcalendar/license.html>; <http://www.edankert.com/bounce/index.html>; <http://www.net-snmp.org/about/license.html>; <http://www.openmdx.org/#FAQ>; http://www.php.net/license/3_01.txt; <http://srp.stanford.edu/license.txt>; <http://www.schneier.com/blowfish.html>; <http://www.jmock.org/license.html>; <http://xsom.java.net>; <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>; <https://github.com/lyokato/libgeohash/blob/master/LICENSE>; <https://github.com/hjiang/jsonxx/blob/master/LICENSE>; <https://code.google.com/p/lz4/>; <https://github.com/jedisct1/libsodium/blob/master/LICENSE>; <http://one-jar.sourceforge.net/index.php?page=documents&file=license>; <https://github.com/EsotericSoftware/kryo/blob/master/license.txt>; <http://www.scalalang.org/license.html>; <https://github.com/tinkerpop/blueprints/blob/master/LICENSE.txt>; and <http://gee.cs.oswego.edu/dl/classes/EDU/oswego/cs/dl/util/concurrent/intro.html>.

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; 8,707,336; 8,886,617 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: 2018-07-19

Table of Contents

Preface	6
Informatica Resources.	6
Informatica My Support Portal.	6
Informatica Documentation.	6
Informatica Product Availability Matrixes.	6
Informatica Web Site.	6
Informatica How-To Library.	7
Informatica Knowledge Base.	7
Informatica Support YouTube Channel.	7
Informatica Marketplace.	7
Informatica Velocity.	7
Informatica Global Customer Support.	7
Chapter 1: Using Command Line Programs	8
Using Command Line Programs Overview.	8
Entering Options and Arguments.	8
Syntax Notation.	9
Help Command.	9
Chapter 2: Configuring Environment Variables	11
Configuring Environment Variables Overview.	11
Setting Environmental Variables on a Local Machine.	11
Setting Environmental Variables on a Remote Machine.	11
Setting JAVA_HOME.	12
Setting RULEPOINT_CMD_HOME.	12
Chapter 3: rpcmd	13
rpcmd Overview.	13
Run rpcmd Commands Using Arguments from the Input File.	14
Running rpcmd Commands from Command Prompt.	15
Setting Up Server Configuration.	15
deploy.	16
deployAll.	17
downloadFile.	18
export.	19
exportAll.	20
importAll.	21
list.	22
listAll.	23
listtopology.	24

reassign.	24
redeploy.	25
redeployAll.	26
runOnce.	27
start.	28
stop.	29
undeploy.	29
undeployAll.	30
uploadFile.	31
Chapter 4: passwordutil.	33
passwordutil Overview.	33
Running passwordUtil.	33
encrypt.	33
decrypt.	34
passwordUtil Help.	34
Index.	35

Preface

The *RulePoint Command Reference* is written for RulePoint administrators and developers who manage and administer the design time and run-time environments. This guide assumes you have knowledge of the operating systems in your environment. This guide also assumes you are familiar with the interface requirements for the supporting applications.

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 makes 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 Product Availability Matrixes

Product Availability Matrixes (PAMs) indicate the versions of operating systems, databases, and other types of data sources and targets that a product release supports. You can access the PAMs on the Informatica My Support Portal at <https://mysupport.informatica.com/community/my-support/product-availability-matrixes>.

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

Using Command Line Programs

This chapter includes the following topics:

- [Using Command Line Programs Overview, 8](#)
- [Entering Options and Arguments, 8](#)
- [Syntax Notation, 9](#)
- [Help Command, 9](#)

Using Command Line Programs Overview

You use the user interface to perform all tasks related to the design-time and the run-time environment. You can also use the command line programs to perform tasks in RulePoint from a local or remote machine.

For example, you can use the command line programs to import and export objects, perform deployment tasks for objects from the design-time environment to the application services in the run-time environment, and administer the run-time environment. The command line programs allow you to perform tasks that you could otherwise complete using only the user interface.

RulePoint includes the following command line programs:

- `rpcmd`. Use `rpcmd` to administer the RulePoint topology and objects.
- `passwordUtil`. Use `passwordUtil` to encrypt or decrypt database passwords.

Entering Options and Arguments

Each command line program requires the URL, user name, password, and command name. You need to specify the rest of the options and arguments in the command line interface based on your requirement.

Use the following rules when you enter command options and arguments:

- To enter options, type a hyphen followed by one letter, two letters, or a word, depending on the program syntax for the command. For example, the `rpcmd < -pn | -ProjectName> project_name` uses a two-letter option or a word for the project name.
- Enter options in any order.
- If any option that you specify from the command line contains spaces, enclose the option in double quotes.

- The first word after the option is the argument.
- Most options require arguments. You must separate options from arguments with a single space.
- If any argument contains more than one word, enclose the argument in double quotes. The command line programs ignore quotes that do not enclose an argument. Unmatched quotes result in an error.

Syntax Notation

The following table describes the notations used for the RulePoint command line program:

Convention	Description
-x	Option placed before a argument. This designates the parameter you enter. For example, to enter the user name for deploy command, type -un or -UserName followed by the user name.
< x >	Required option. If you omit a required option, the command line program returns an error message.
[x]	Optional parameter. The command runs whether or not you enter optional parameters. For example, the Help command has the following syntax: help [COMMAND]

Help Command

You can use the help to list all the commands. If you enter the command name along with the help, rpcmd lists all options and arguments for that command.

The Help command uses the following syntax:

```
help [COMMAND]
```

For example, if you type rpcmd help, rpcmd returns the following options and arguments:

```
USAGE: rpcmd configure  Configure the properties required to connect to the RulePoint
server
      rpcmd COMMAND <-u | -URL> url <-un | -UserName> username <-pd | -Password>
password <-f | -File> file_name
      rpcmd COMMAND <-u | -URL> url <-un | -UserName> username <-pd | -Password>
password [options]
      rpcmd help [COMMAND]
where COMMAND can be one of the following:
deploy          Deploy one or more objects of a specific type in a given RulePoint
Project.
deployAll      Deploy all objects or objects of a specific type in a given RulePoint
Project.
undeploy       Undeploy one or more objects of a specific type in a given RulePoint
Project.
undeployAll    Undeploy all objects or objects of a specific type in a given RulePoint
Project.
redeploy       Redeploy one or more objects of a specific type in a given RulePoint
Project.
redeployAll    Redeploy all objects or objects of a specific type in a given RulePoint
Project.
```

```

reassign      Reassign a ApplicationService to one or more RulePoint objects within a
project.
list         List all the objects within a RulePoint Project of a given type.
listtopology List the topology of the RulePoint setup.
runOnce      Run the deployed source once. It is applicable only for schedulable
sources.
start        Start a deployed object which is already in stopped/deployed state to
poll data, generate events or generating responses.
stop         Stop a deployed object which is in started state, from polling data,
generating events or generating responses.
uploadFile   Upload File to RulePoint Server.
importAll    Import all objects from an uploaded XML file into a specific RulePoint
project.
listAll      List all the XML files that are currently on the RulePoint Server.
exportAll    Export all objects from a RulePoint project into XML File.
export       Export selected objects from a RulePoint project into XML File.
downloadFile Download a File from RulePoint Server to local hard drive.

```

The following table describes the rpcmd help option and argument:

Option	Argument	Description
Not applicable	command	Optional. Name of command. If you omit the command name, rpcmd lists all commands.

CHAPTER 2

Configuring Environment Variables

This chapter includes the following topics:

- [Configuring Environment Variables Overview, 11](#)
- [Setting Environmental Variables on a Local Machine, 11](#)
- [Setting Environmental Variables on a Remote Machine, 11](#)

Configuring Environment Variables Overview

You can run the `rpcmd` command from a local machine where you have installed RulePoint or from a remote machine.

Setting Environmental Variables on a Local Machine

Verify that you have set `RULEPOINT_HOME` to the location of the RulePoint installation directory.

Extract the contents from the `rpcmd.zip` file into the `<RulePoint installation Directory>\tools\rpcmd` directory.

Setting Environmental Variables on a Remote Machine

When you use a remote machine, you need to set the environmental variables, `JAVA_HOME` and `RULEPOINT_CMD_HOME` on the remote machine. Copy the `rpcmd.zip` file from the installation directory to a directory on your machine where you want to run the command, and then extract the contents of the `rpcmd.zip` files.

Setting JAVA_HOME

Set JAVA_HOME to the location of the Java installation directory.

To configure JAVA_HOME on Linux, type the following command based on your environment:

- Linux C shell: `setenv JAVA_HOME <Java installation directory>`
- Linux Bourne shell: `export JAVA_HOME = <Java installation directory>`

To configure JAVA_HOME on Windows, set the JAVA_HOME environmental variable to point to *Java Installation directory*. You can also set JAVA_HOME in the system settings.

Setting RULEPOINT_CMD_HOME

Create the RULEPOINT_CMD_HOME environment variable.

To configure RULEPOINT_CMD_HOME on Linux, run one of the following commands based on your environment:

- Linux C shell: `setenv RULEPOINT_CMD_HOME <unzipped rpcmd directory>`
- Linux Bourne shell: `export RULEPOINT_CMD_HOME = <unzipped rpcmd directory>`

To configure RULEPOINT_CMD_HOME on Windows, set the RULEPOINT_CMD_HOME environmental variable to point to *unzipped rpcmd directory*. You can also set RULEPOINT_CMD_HOME in the system settings.

CHAPTER 3

rpcmd

This chapter includes the following topics:

- [rpcmd Overview, 13](#)
- [Run rpcmd Commands Using Arguments from the Input File, 14](#)
- [Running rpcmd Commands from Command Prompt, 15](#)
- [Setting Up Server Configuration, 15](#)
- [deploy, 16](#)
- [deployAll, 17](#)
- [downloadFile, 18](#)
- [export, 19](#)
- [exportAll, 20](#)
- [importAll, 21](#)
- [list, 22](#)
- [listAll, 23](#)
- [listtopology, 24](#)
- [reassign, 24](#)
- [redeploy, 25](#)
- [redeployAll, 26](#)
- [runOnce, 27](#)
- [start, 28](#)
- [stop, 29](#)
- [undeploy, 29](#)
- [undeployAll, 30](#)
- [uploadFile, 31](#)

rpcmd Overview

rpcmd is a command line program that you can use to administer the RulePoint topology and objects.

Use rpcmd to administer the following tasks for objects and services:

- Deploy one or more primary objects from the design-time environment to run-time environment.

- Undeploy or redeploy one or more primary objects.
- Reassign all primary objects from one service controller to another service controller.
- Get a list of all primary objects with information about its deployed state.
- Get information about the topology and the services configured in the topology.
- Export a backup of the current working project to a file system.
- Import objects into a specific project.
- Perform dashboard tasks, such as run once, stop, or start.

You invoke `rpcmd` from the command line. You can issue commands directly or from a script, batch file, or other program.

Run `rpcmd` Commands Using Arguments from the Input File

You can create a `.txt` file to call the `rpcmd` commands including its options and arguments, and then run the script in the command line.

1. To perform a task, create a text file by adding all the parameters you want to perform that particular task.

The file must contain the URL, username, password, and the required values.

2. Place the text file in the RulePoint installation folder at: `<RULEPOINT_CMD_HOME>/scripts`.
3. To call the text file, run the following script:

```
rpcmd COMMAND <-f | -File> file_name
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
```

where:

- `COMMAND` is the name of the command for a specific task.
- `file_name` is name of the file where you have specified the command parameters.

Example

To perform the Deploy function by using the input file, you must first create a file, `deploy.txt`, and add the following parameters that are necessary for deployment:

```
ProjectName=Banking_UC
ObjectType=sources
ObjectName=Customer Accounts Source
ApplicationServiceName=Default Source Controller
TopologyName=Default
```

Run the script from the command line:

```
rpcmd.bat deploy -f deploy.txt -u http://localhost:8080/rulepoint -un Administrator -pd Administrator1
```

Running rpcmd Commands from Command Prompt

1. At the command prompt, switch to the directory where the rpcmd executable is located.
By default, rpcmd installs in the `RULEPOINT_CMD_HOME\bin` directory.
2. Run the following command, followed by the command name and its required options and arguments:
 - On Windows, run `rpcmd.bat`
 - On Linux, run `rpcmd.sh`

Example

```
rpcmd <-URL|-u> url <-UserName|-un>  
user_name <-Password|-pd> password
```

When you run rpcmd, you enter options for each command, followed by the required arguments. For example, most commands require that you enter the user name, password, and command name using the command options. Command options are preceded with a hyphen and are case sensitive. Arguments follow the option. To enter an argument that contains a space or other non-alphanumeric character, enclose the argument in quotation marks.

For example, to deploy all the sources in a specific project on a particular source controller on Windows:

```
rpcmd.bat deployAll -u http://localhost:8080/rulepoint -un Administrator -pd  
Administrator1 -pn "Default Project" -ot sources -as "Default Source Controller"
```

On Linux, run the following command:

```
rpcmd.sh deployAll -u http://localhost:8080/rulepoint -un Administrator -pd  
Administrator1 -pn "Default Project" -ot sources -as "Default Source Controller"
```

If you omit or incorrectly enter one of the required options, the command fails and rpcmd returns an error message.

Setting Up Server Configuration

Configure the connection settings for the RulePoint server if you want to use https to run rpcmd. Use the `configure` command to set up https or http for all the commands.

The command uses the following syntax:

```
rpcmd.sh configure
```

Make sure that you have configured secure connection during RulePoint installation.

1. Run the following command to configure the properties to connect to the RulePoint server
`configure`
2. The utility prompts you if you want to enable HTTPS.
3. To enable HTTPS, type **Y** and press **ENTER**.
4. If you do not want to enable HTTPS, type **N**, and skip to Step 6.
5. If you type **Y**, perform the following tasks:
 - a. To generate the keystore file, type **N**. The file is generated in the `RULEPOINT_CMD_HOME/bin` directory.
 - b. To use your own keystore file, type **Y**, and press **ENTER**.
 - c. Enter the keystore path, and press **ENTER**.

- d. Enter the password for the keystore file, if prompted.
6. Enter the host name or IP address of the machine where RulePoint is installed.
7. Enter the port number. The port must be unique. The default port number is 8443 for https and 8080 for http.
8. Enter the application service name. By default, the service name is rulepoint.
9. Enter the user name and password that you use for the RulePoint login.

You can use either the http or the https connection to connect to the RulePoint server . When you plan to use https, configure the https settings and run the commands for the various tasks directly. You can choose to override the secure settings while running the commands. In this case, you need to add the connection settings.

Example

If you configured the connections settings, use the following syntax to deploy all objects:

```
/rpcmd.sh deployall -pn Banking -at responders -as "Default Responder Controller"
```

If you want to override the connection settings, use the following syntax to deploy all objects:

```
/rpcmd.sh deployall -u http://localhost:8080/rulepoint -un Administrator -pd Administrator1 -pn Banking -at responders -as "Default Responder Controller"
```

deploy

Deploys one or more primary objects from the design-time environment to the application services in the run-time environment. If you want to run the CLI command for deployment, you must have the Execute or Administrator privileges for that object.

The deploy command uses the following syntax:

```
rpcmd deploy
  <-URL|-u> url
  <-UserName|-un> user_name
  <-Password|-pd> password
  <-ProjectName|-pn> project_name
  <-ObjectType|-ot> object_type
  <-ObjectName|-on> object_name1, object_name2...| file:file_name
  [<-ApplicationServiceName|-as> application_service]
  [<-TopologyName|-tn> topology_name]
```

The following table describes the rpcmd deploy options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.

Option	Argument	Description
-ProjectName -pn	project_name	The name of the RulePoint project which contains the object.
-ObjectName -on	object_name1	Name of the RulePoint object that you want to deploy. To enter more than one object, separate them with a comma. Alternatively, you can also enter the object names from an input file located at <RulePoint installation Directory>/scripts.
-ApplicationServiceName -as	application_service	Name of the source controller, responder controller, or event processor. If there is only a single instance of the application service, this field is optional. For example, if there is only one event processor in the system while you deploy a rule, you can omit the name of the event processor.
-ObjectType -ot	object_type	Type of object that you want to deploy. Valid values for this option are sources, rules, and responders
-TopologyName -tn	topology_name	Optional. Name of the topology.

Example

You can use the following syntax to deploy the banking use case sources to a source controller:

```
> ./rpcmd.sh deploy -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn Banking_UC -ot sources -on "Customer Accounts Source" -as "Default
Source Controller" -tn Default
```

Use the following syntax to reference the object names from an input file:

```
> ./rpcmd.sh deploy -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn Banking_UC -on file:Banking_UC.txt -ot sources -as "Default Source
Controller" -tn Default
```

deployAll

Deploys all objects in a given RulePoint project. You can run this command on a single source controller, responder controller, or event processor.

The deployAll command uses the following syntax:

```
rpcmd deployAll
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
<-ProjectName|-pn> project_name
[<-ObjectType|-at> object_type]
[<-ApplicationServiceName|-as> application_service]
[<-TopologyName|-tn> topology_name]
```

The following table describes the rpcmd deployAll options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-ProjectName -pn	project_name	The name of the RulePoint project which contains the object.
-ObjectName -on	object_name1	Name of the RulePoint objects that you want to deploy. To enter more than one object, separate them with a comma.
-ApplicationServiceName -as	application_service	Name of the source controller, responder controller, or event processor. If there is only a single instance of the application service, this field is optional. For example, if there is only one event processor in the system while you deploy a rule, you can omit the name of the event processor.
-ObjectType -ot	object_type	Type of objects that you want to deploy. Valid values for this option are sources, rules, and responders
-TopologyName -tn	topology_name	Optional. Name of the topology.

Example

You can use the following syntax to deploy all sources from the banking use case project:

```
> ./rpcmd.sh deployall -u http://localhost:8080/rulepoint -un Administrator -pd  
Administrator1 -pn "Default Project" -ot sources
```

downloadFile

Downloads a file from the RulePoint server to the local system.

The downloadFile command uses the following syntax:

```
rpcmd downloadFile  
<-URL|-u> url  
<-UserName|-un> user_name  
<-Password|-pd> password  
<-FileName|-fn> file_name
```

The following table describes the rpcmd downloadFile options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-FileName -fn	file_name	Name of XML file present in the RulePoint server.

Example

You can run the following command to download a file:

```
> ./rpcmd.sh downloadFile -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -fn export_DP_Objects_1382523536662.xml
```

export

Exports selected objects to an XML file.

The export command uses the following syntax:

```
rpcmd export
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
<-ProjectName|-pn> project_name
-ObjectName|-on <object_name1, object_name2, ... | file:file_name>
-ObjectType|-ot <Object_type>
-fn <file_name>
[-IncludeAllDependencies|-iad <include_all_dependencies>]
[-IncludeAdditionalExportOptions|-ixp <ixp>]
[-Users|-us <user_name1, user_name2, ... | file:file_name>]
<-FileName|-fn> file_name
```

The following table describes the rpcmd exportSelected options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.

Option	Argument	Description
-ProjectName -pn	project_name	Name of RulePoint project from where you want to export selected objects.
-ObjectName -on	object_name1, object_name2...	List of one or more RulePoint object names for which you want to run the command. You can provide the list of object names from a file.
-ObjectType -ot	object_type	Type of object that you want to export. You can provide one of the following options: sources, responders, rules, topics, responses, analytics, watchlists, connections, or templates.
-Users -us	user_name	The users in the system that you want to export.
-FileName -fn	file_name	The XML filename to which you want to export the objects.
-IncludeAllDependencies -iad	include_all_dependencies	To include all supporting objects, set iad value as true.
-IncludeAdditionalExportOptions -ixp	include_export_options	Optional. Additional options for exporting data. You can use the following options: <ul style="list-style-type: none"> - includeACLs. Exports all ACLs. - includeInvalidObjects. Exports all ACLs invalid objects that are present in the system. - includeSystemConfiguration. export the system configuration information along with objects.

Example

You can run the following command to export rules from the project "Banking" to the

Export_banking_Objects.xml file:

```
./rpcmd.bat export -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn Banking -on "Rule to check for amount withdrawn that exceeds the
limit,Rule to check for customers living in a city where health checkup is organized" -
ot rules -fn Export_banking_Objects
```

exportAll

Exports all objects from a RulePoint project into an XML file.

The exportAll command uses the following syntax:

```
rpcmd exportAll
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
<-ProjectName|-pn> project_name
<-FileName|-fn> file_name
[<-IncludeAdditionalExportOptions|-ixp> include_export_options]
```

The following table describes the rpcmd exportAll options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-ProjectName -pn	project_name	Name of RulePoint project from where you want to export all objects.
-FileName -fn	file_name	The name of file that will be created with all the objects from the RulePoint project.
IncludeAdditionalExportOptions -ixp	include_export_options	Optional. Additional options for exporting data. You can use the following options: <ul style="list-style-type: none"> - includeACLs. Exports all ACLs. - includeInvalidObjects. Exports all ACLs invalid objects that are present in the system. - includeSystemConfiguration. export the system configuration information along with objects.

Example

You can run the following command to export all objects:

```
> ./rpcmd.sh exportall -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn "Default Project" -fn export_objects -ixp includeACL
```

importAll

Imports all objects from an uploaded XML file into a specific RulePoint project.

Note: When you import objects, verify that the file contains all the supporting objects.

The uploadFile command uses the following syntax:

```
rpcmd importAll
  <-URL|-u> url
  <-UserName|-un> user_name
  <-Password|-pd> password
  <-ProjectName|-pn> project_name
  <-FileName|-fn> file_name
  [<-ImportMode|-im> import_mode]
```

The following table describes the rpcmd importAll options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-ProjectName -pn	project_name	Name of RulePoint project where you want to import the objects.
-FileName -fn	file_name	The XML filename that you have uploaded to the RulePoint server. The file contains the objects that you want to import.
-ImportMode -im	import_mode	The mode that you can specify for importing objects, so that the objects can either fail, skip, or update existing objects in the project.

Example

You can run the following command to import all objects from an uploaded XML file, `Banking_Sample.xml`, from the project "Default Project." The import mode used is "UPDATE."

```
> ./rpcmd.sh importall -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn "Default Project" -fn Banking_Sample.xml -im UPDATE
```

list

Use this command to list objects of a specific type within a RulePoint project. For example, you can fetch a list of sources, rules, or responders.

The list command uses the following syntax:

```
rpcmd list
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
<-ProjectName|-pn> project_name
[<-ObjectType|-at> object_type]
```

The following table describes the rpcmd list options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.

Option	Argument	Description
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-ProjectName -pn	project_name	The name of the RulePoint project where the object is present.
-ObjectType -ot	object_type	Type of object that you want to redeploy. Valid values for this option are sources, rules, and responders. The command lists all the available objects for the specified value, along with the object type, name, validity, and deployment state (Deployed, Needs_Deployment, or Draft).

Example

You can run the following command to list all the objects, along with information for each object type:

```
> ./rpcmd.sh list -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn "Default Project"
```

listAll

Lists all the XML files that are in the RulePoint Server.

The listAll command uses the following syntax:

```
rpcmd listAll
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
```

The following table describes the rpcmd listAll options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.

Example

You can use the listAll command to list all objects in a project:

```
> ./rpcmd.sh listall -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1
```

listtopology

Lists the topology components of the RulePoint setup.

The listtopology command uses the following syntax:

```
rpcmd listtopology
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
[<-TopologyName|-tn> topology_name]
```

The following table describes the rpcmd listtopology options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-TopologyName -tn	topology_name	The name of the topology defined in RulePoint.

Example

You can run the following command to list the topology elements:

```
> ./rpcmd.sh listtopology -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -tn Default
```

reassign

Reassigns one or more RulePoint objects to an application service within a project.

The reassign command uses the following syntax:

```
rpcmd reassign
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
<-ProjectName|-pn> project_name
<-ObjectType|-ot> Object_type
<-ObjectName|-on> object_name1, object_name2...
<-ApplicationServiceName|-as> application_service
[<-TopologyName|-tn> topology_name]
```


The following table describes the rpcmd reassign options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-ProjectName -pn	project_name	Name of RulePoint project where the object is present.
-ObjectName -on	object_name1	Name of the RulePoint object that you want to reassign. To enter multiple objects, separate them with a comma. Alternatively, you can also enter the object names from an input file located at <RulePoint installation Directory>/scripts.
-ApplicationServiceName -as	application_service	Name of the source controller, responder controller, or event processor to which you want to reassign the object.
-ObjectType -ot	object_type	Type of object that you want to reassign. Valid values for this option are sources, rules, and responders.
-TopologyName -tn	topology_name	Optional. Name of the topology.

Example

You can use the following syntax to reassign all sources from one source controller to another source controller:

```
> ./rpcmd.sh reassign -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn "Default Project" -ot sources -on sql_src -as srcl -tn Default
```

redeploy

Redeploys one or more objects. You can redeploy objects that are in the Needs_Deployment state.

The redeploy command uses the following syntax:

```
rpcmd redeploy
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
<-ProjectName|-pn> project_name
<-ObjectName|-on> object_name1, object_name2...| file:file_name
[<-ObjectType|-ot> object_type]
```

The following table describes the rpcmd redeploy options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-ProjectName -pn	project_name	The name of the RulePoint project where the object is present.
-ObjectName -on	object_name1	The name of the RulePoint object that you want to redeploy. To enter multiple objects, separate them with a comma. Alternatively, you can also enter the object names from an input file located at <RulePoint installation Directory>/scripts.
-ObjectType -ot	object_type	Type of object that you want to redeploy. Valid values for this option are sources, rules, and responders.

Example

You can use the following syntax to redeploy multiple rules from the banking usecase project:

```
> ./rpcmd.sh redeploy -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn Banking_UC -ot sources -on "Customer Accounts Source"
```

redeployAll

Redeploys all objects in a given RulePoint project.

The redeployAll command uses the following syntax:

```
rpcmd redeployAll
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
<-ProjectName|-pn> project_name
[<-ObjectType|-ot> object_type]
```

The following table describes the rpcmd redeployAll options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.

Option	Argument	Description
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-ProjectName -pn	project_name	The name of the RulePoint project where the object is present.
-ObjectType -ot	object_type	Type of object that you want to redeploy. Valid values for this option are sources, rules, and responders.

Example

You can use the following syntax to redeploy all sources from the banking usecase project:

```
> ./rpcmd.sh redeployall -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn "Default Project" -ot sources
```

runOnce

You can use the runOnce command for scheduled sources.

Runs the deployed source once, so that the source polls the data only one time and publishes them as events.

The runOnce command uses the following syntax:

```
rpcmd runOnce
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
-ObjectName|-on <object_name>
-ProjectName|-pn <project_name>
[-TopologyName|-tn <topology_name>]
```

The following table describes the rpcmd runOnce options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-ProjectName -pn	project_name	The name of the RulePoint project where the object is present.
-ObjectName -on	object_name	Name of deployed source that you want to run to generate events.

Option	Argument	Description
-ObjectType -ot	object_type	Type of objects that you want to run once. Valid values for this option are sources, rules, and responders.
-TopologyName -tn	topology_name	Optional. Name of the topology.

Example

You can run the following command to run the source once:

```
> ./rpcmd.sh runonce -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn Banking_UC -on "Customer Accounts Source"
```

start

Starts the deployed objects to poll data, generate events, or responses.

The startSource command uses the following syntax:

```
rpcmd start
  <-URL|-u> url
  <-UserName|-un> user_name
  <-Password|-pd> password
    <-ProjectName|-pn> project_name
    <-ObjectName|-an> object_name
  <-ObjectType|-at> object_type
```

The following table describes the rpcmd startSource options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-ProjectName -pn	project_name	The name of the RulePoint project where the object is present.
-ObjectName -on	object_name	Name of the deployed source that you want to start polling data.
-ObjectType -ot	object_type	Type of object that you want to start. Valid values for this option are sources, rules, and responders.

Example

You can run the following command to start the source from the banking use case project:

```
> ./rpcmd.sh start -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn "Default Project" -ot sources -on cust_src
```

stop

Stops the source from polling data and generating events.

The stop command uses the following syntax:

```
rpcmd stop
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
<-ProjectName|-pn> project_name
<-ObjectName|-an> object_name
<-ObjectType|-at> object_type
```

The following table describes the rpcmd stop options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-ProjectName -pn	project_name	The name of the RulePoint project where the object is present.
-ObjectName -on	object_name	Name of the source that you want to stop from polling data.
-ObjectType -ot	object_type	Type of object that you want to stop. Valid values for this option are sources, rules, and responders.

Example

You can run the following command to stop a source from polling data:

```
> ./rpcmd.sh stop -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn "Default Project" -ot sources -on cust_src
```

undeploy

Undeploys one or more objects.

The undeploy command uses the following syntax:

```
rpcmd undeploy
  <-URL|-u> url
  <-UserName|-un> user_name
  <-Password|-pd> password
  <-ProjectName|-pn> project_name
  <-ObjectName|-an> object_name1, object_name2...|file;file_name
  [<-ObjectType|-at> object_type]
```

The following table describes the rpcmd undeploy options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-ProjectName -pn	project_name	Name of RulePoint project that contains the object.
-ObjectName -on	object_name1	The name of the RulePoint object that you want to undeploy. If you want to undeploy a number of objects, enter the object names separated with a comma. Alternatively, you can also enter the object names from an input file located at <RulePoint installation Directory>/scripts.
-ObjectType -ot	object_type	Type of object that you want to undeploy. Valid values for this option are sources, rules, and responders.

Example

You can use the following syntax to undeploy multiple rules:

```
> ./rpcmd.sh undeploy -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn Banking_UC -ot rules -on "Template to check for customers with a
given balance_rule","Rule to demonstrate usage of Watchlist and SQL Analytic","Rule to
check for customers with balance below 10000"
```

undeployAll

Undeploys all objects in a given RulePoint project.

The undeployAll command uses the following syntax:

```
rpcmd undeployAll
  <-URL|-u> url
  <-UserName|-un> user_name
  <-Password|-pd> password
  <-ProjectName|-pn> project_name
  [<-ObjectType|-at> object_type]
```

The following table describes the rpcmd undeployAll options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-ProjectName -pn	project_name	Name of RulePoint project that contains the object.
-ObjectType -ot	Object_type	Type of object that you want to undeploy. Valid values for this option are sources, rules, and responders.

Example

You can use the following syntax to undeploy all sources:

```
> ./rpcmd.sh undeployall -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn "Default Project" -ot sources
```

uploadFile

Uploads a file to the RulePoint server

The uploadFile command uses the following syntax:

```
rpcmd uploadFile
<-URL|-u> url
<-UserName|-un> user_name
<-Password|-pd> password
<-FileName|-fn> file_name
```

The following table describes the rpcmd uploadFile options and arguments:

Option	Argument	Description
-URL -u	url	URL of RulePoint design server instance to connect to run the command.
-UserName -un	user_name	User name to connect to RulePoint Server.

Option	Argument	Description
-Password -pd	password	Password of the user name to connect to the RulePoint Server.
-FileName -fn	file_name	The file path to the XML file that contains the objects.

Example

You can run the following command to upload the file `BankingObjects.xml` as part of the project named *Default Project*.

```
> ./rpcmd.sh uploadFile -u http://localhost:8080/rulepoint -un Administrator -pd
Administrator1 -pn "Default Project" -fn /data/home/cepqa/RulePoint_6.1.2/samples/
Banking/db/Oracle/Banking_Sample.xml
```


CHAPTER 4

passwordutil

This chapter includes the following topics:

- [passwordutil Overview, 33](#)
- [Running passwordUtil, 33](#)

passwordutil Overview

You can use the passwordUtil to encrypt or decrypt database passwords

If the password has special characters, you must perform the following tasks based on your environment:

- On Windows, enclose the special characters with double quotes. For example, "system&b"
- On Linux, enclose the special characters with the escape character. For example, \ (example : system \ `b)

Running passwordUtil

To run the passwordUtil command directly from the command line interface:

1. At the command prompt, switch to the directory where the passwordutil executable is located.
By default, passwordUtil installs in the `RULEPOINT_CMD_HOME\bin` directory.
2. Run the following command, followed by the command name and its required options and arguments, based on your environment:
 - Windows: passwordUtil.bat
 - Linux: passwordUtil.shThe command names are not case sensitive.

encrypt

Encrypts a password.

Use the following syntax to encrypt a password:

```
passwordUtil.sh encrypt <password in plain text format>
```

When you run `passwordUtil`, you enter options for each command, followed by the required arguments. To enter an argument that contains a space or other non-alphanumeric character, enclose the argument in quotation marks.

For example, to encrypt the password for a particular database on Windows:

```
passwordUtil.bat encrypt system123
```

On Linux, run the following command:

```
passwordUtil.sh encrypt system123
```

If you omit or incorrectly enter one of the required options, the command fails and `passwordUtil` returns an error message.

The `passwordUtil` command returns the following value:

```
25a0bdfd1ec9e4f6a648097acc5acfe4e0240a65fbaad820d91e0c72886484
```

decrypt

Decrypts a password.

Use the following syntax to decrypt a password:

```
passwordUtil.sh decrypt <password in encrypted format>
```

For example, to decrypt the password for a particular database on Windows:

```
passwordUtil.bat decrypt 25a0bdfd1ec9e4f6a648097acc5acfe4e0240a65fbaad820d91e0c72886484
```

On Linux, run the following command:

```
passwordUtil.sh decrypt 25a0bdfd1ec9e4f6a648097acc5acfe4e0240a65fbaad820d91e0c72886484
```

If you omit or incorrectly enter one of the required options, the command fails and `passwordUtil` returns an error.

The `passwordUtil` command returns the following value:

```
system123
```

passwordUtil Help

You can use the help to list the encrypt and decrypt commands. `passwordUtil` lists all options and arguments for that command.

For example, if you type `passwordUtil help`, `passwordUtil` returns the following options and arguments:

```
Usage: PasswordUtil <encrypt | decrypt> input_string
encrypt          Encrypt the given input string.
decrypt         Decrypt the given input string.
```

INDEX

C

- command line program
 - overview [8](#)
- command line programs
 - syntax for [9](#)
- command syntax
 - deploy [16](#)
 - deployAll [17](#)
 - downloadFile [18](#)
 - export [19](#)
 - exportAll [20](#)
 - importAll [21](#)
 - list [22](#)
 - listAll [23](#)
 - listtopology [24](#)
 - reassign [24](#)
 - redploy [25](#)
 - redployAll [26](#)
 - runOnce [27](#)
 - start [28](#)
 - stop [29](#)
 - undeploy [29](#)
 - undeployAll [30](#)
 - uploadFile [31](#)
- commands
 - entering options and arguments for [8](#)

E

- environmental variable
 - setting JAVA_HOME [12](#)

- environmental variable (*continued*)
 - setting RULEPOINT_CMD_HOME [12](#)

O

- options and arguments
 - deploy [16](#)
 - deployAll [17](#)
 - downloadFile [18](#)
 - export [19](#)
 - exportAll [20](#)
 - importAll [21](#)
 - list [22](#)
 - listAll [23](#)
 - listtopology [24](#)
 - reassign [24](#)
 - redploy [25](#)
 - redployAll [26](#)
 - runOnce [27](#)
 - start [28](#)
 - stop [29](#)
 - undeploy [29](#)
 - undeployAll [30](#)
 - uploadFile [31](#)

S

- syntax
 - command line programs [9](#)