

Rational Publishing Engine

API Reference

Release 1.1

Before using this information, be sure to read the general information under Appendix, [“Notices” on page 11.](#)

This edition applies to **VERSION 1.1, Rational Publishing Engine** and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2009**

US Government Users Restricted Rights—Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Table of Contents

Programmatic Control for IBM Rational Publishing Engine	1
RPE API	1
Requirements	1
Configuration Files	1
Using the API for Publishing Reports with the Local RPE Installation	2
Requirements	2
Example	2
Using the API for Publishing Reports with a Remote RPE Web Service	4
Requirements	4
Example	4
Web Service Definition	6
Operations	6
Programmatic Control for RPE Web Service Without Using the RPE API	6
Command Line Switches for rpe-launcher.exe	9
Appendix A: Notices	11
Copyright license	13
Trademarks	14

Programmatic Control for IBM Rational Publishing Engine

RPE API

Starting with Rational Publishing Engine 1.1, a programmatic interface is made public for client applications to control the publishing process. The API is documented using JavaDoc.

RPE can publish documents in two modes: local and remote. For more details regarding these modes of operation, refer to sections [Using the API for Publishing Reports with the Local RPE Installation Page 2](#) and [Using the API for Publishing Reports with a Remote RPE Web Service Page 4](#), respectively.

Requirements

The API is not dependant on the operating system, however it is only supported on platforms supported by RPE.

Required files are specific to the operating system and the mode of operation. Refer to sections [Requirements Page 2](#) and [Requirements Page 4](#) for more details.

Configuration Files

Both LocalDocumentGenerator and RemoteDocumentGenerator utility classes require a RPEDocumentSpecification object for publishing a Document Specification.

A RPEDocumentSpecification object is constructed using the com.ibm.rational.rpe.api.docspec.DocumentSpecificationBuilder class, which requires the path to the RPE configuration file.

In case the client application runs on a machine where RPE is installed, the path to the RPE configuration file can be obtained as follows:

```
String configPath = docGen.getConfigurationPath(null);
```

If the client application runs on a machine where RPE is not installed, the path to the RPE configuration file is obtained by passing the path to the folder containing the "config" and "source" subfolders:

```
File configFolder = new File("path/to/folder");  
String configPath = docGen.getConfigurationPath(configFolder.toURI());
```

Where docGen is an object of type LocalDocumentGenerator or RemoteDocumentGenerator.

Using the API for Publishing Reports with the Local RPE Installation

Requirements

The class `com.ibm.rational.rpe.api.docgen.LocalDocumentGenerator` is needed for publishing reports using the local RPE installation.

The files required for this usage scenario are listed in [Table 1 Required files for local publishing. The unique entries are bolded.](#) These files are available in the "%RPE_HOME%\lib" folder of the RPE installation.

Windows	Linux
Aspose.Pdf.jar	Aspose.Pdf.jar
Aspose.Words.jdk16.jar	Aspose.Words.jdk16.jar
commons-lang-2.3.jar	commons-lang-2.3.jar
doors_interop.jar	libtl_lic.so
log4j-1.2.13.jar	log4j-1.2.13.jar
org.eclipse.emf.common_2.4.0.v200808251517.jar	org.eclipse.emf.common_2.4.0.v200808251517.jar
org.eclipse.emf.ecore_2.4.1.v200808251517.jar	org.eclipse.emf.ecore_2.4.1.v200808251517.jar
org.eclipse.emf_2.4.0.v200808251517.jar	org.eclipse.emf_2.4.0.v200808251517.jar
org.eclipse.xsd_2.4.1.v200808251517.jar	org.eclipse.xsd_2.4.1.v200808251517.jar
rpe-common.jar	rpe-common.jar
rpe-engine.jar	rpe-engine.jar
tauaccess.jar (Windows version)	tauaccess.jar (Linux version)
tl_lic.dll	tl_lic.jar
tl_lic.jar	

Table 1 Required files for local publishing. The unique entries are bolded.

Example

The following sample code is an excerpt from the 'rpe_sample_local' java project that shows how to publish a document.

```
LocalDocumentGenerator docGen = new LocalDocumentGenerator();
docGen.addStatusListener(new SampleListener());
```

`try`


```
{
    String config = docGen.getConfigurationPath(null);
    RPEDocumentSpecification docSpec = constructDocSpec(config);
    docGen.publishSync(docSpec);
}
catch (RPEException e)
{
    e.printStackTrace();
}
```

For more information regarding using the API to publish reports using the local RPE installation, see the "rpe_sample_local" java project located in the 'manuals' sub-folder in %RPE_HOME%.

Using the API for Publishing Reports with a Remote RPE Web Service

Requirements

The class `com.ibm.rational.rpe.api.docgen.RemoteDocumentGenerator` is needed for publishing reports using a remote RPE Web Service.

The files required for this usage scenario are listed in [Table 2 Required files for remote publishing](#). These files are available in the "%RPE_HOME%\lib" folder of the RPE installation. Axis2 can be downloaded from <http://ws.apache.org/axis2/download.cgi>.

Windows and Linux
Axis2 1.4 or later
log4j-1.2.13.jar org.eclipse.emf.common_2.4.0.v200808251517.jar org.eclipse.emf.ecore_2.4.1.v200808251517.jar org.eclipse.xsd_2.4.1.v200808251517.jar rpe-common.jar rpe-engine.jar

Table 2 Required files for remote publishing

Example

The following sample code is an excerpt from the 'rpe_sample_remote' project that shows how to publish a document.

```
RemoteDocumentGenerator docGen = new
RemoteDocumentGenerator(SERVER_ADDRESS);
docGen.addStatusListener(new SampleListener());

try
{
    String config = docGen.getConfigurationPath(null);
    RPEDocumentSpecification docSpec = constructDocSpec(config);
    docGen.publishSync(docSpec);
}
catch (RPEException e)
```

```
{  
    e.printStackTrace();  
}
```

```
URI result = docGen.getResultArchive();  
System.out.println("URL of the result archive: " + result);
```

For more information regarding using the API to publish reports using a remote RPE Web Service, see the "rpe_sample_remote" java project located in the 'manuals' sub-folder in %RPE_HOME%.

Web Service Definition

Operations

The four operations available in the RPE Web Service are described in [Table 3 RPE WS operations](#).

Operation Name	Parameter(s)	Return Values
runReportFile	attachment { fileName : string file : base64binary } 	reportID : string
stopReport	reportID : integer	confirmation : string
getStatus	reportID : integer	statusResponse { message: string engineStatus : integer }
getOutput	reportID : integer	output URL : string

Table 3 RPE WS operations

Programmatic Control for RPE Web Service Without Using the RPE API

Using the WSDL file from the `http://server:port/rpe/services/RPEService?wsdl` address, a top-down Web Service Client application can be generated. The following sample code demonstrates how an existing Document Specification file can be published remotely, without using the RPE API.

```
StatusResponse statusResponse = null;
Integer serviceRunResponse;
RPEServiceStub stub = new RPEServiceStub("http://localhost:8080/rpe/
services/RPEService?wsdl");
stub._getServiceClient().getOptions().setProperty(Constants.Configuration
n.ENABLE_MTOM, Constants.VALUE_TRUE);
stub._getServiceClient().getOptions().setTimeoutInMilliseconds(10000);
```

```
RunReportFile request = (RunReportFile)
RunReportFile.class.newInstance();
AttachmentType attachment = new AttachmentType();
Base64Binary base64Binary = new Base64Binary();

File docSpecFile = new File("c:/docSpec.dsx");

FileDataSource fileDataSource = new FileDataSource(docSpecFile);
DataHandler dataHandler = new DataHandler(fileDataSource);
base64Binary.setBase64Binary(dataHandler);

ContentType_type0 param = new ContentType_type0();
param.setContentType_type0(dataHandler.getContentType());
base64Binary.setContentType(param);
attachment.setFileName(docSpecFile.toURI().toString());
attachment.setBinaryData(base64Binary);
request.setRunReportFile(attachment);
serviceRunResponse =
Integer.valueOf(stub.runReportFile(request).getRunReportFileResponse());

RPEServiceStub.GetStatus getStatus = (RPEServiceStub.GetStatus)
RPEServiceStub.GetStatus.class.newInstance();
    getStatus.setGetStatus(serviceRunResponse);
    statusResponse =
stub.getStatus(getStatus).getGetStatusResponse();

// 2 = "engine running" status
while ((statusResponse.getEngineStatus() == 2) ||
(!statusResponse.getMessage().equalsIgnoreCase("no message")))
{
    Thread.sleep(1000);
    if (!statusResponse.getMessage().equalsIgnoreCase("no
message"))
    {
        System.out.println(statusResponse.getMessage());
    }
}
```

```
        statusResponse =
stub.getStatus(getStatus).getGetStatusResponse();
    }

    // 3 = "engine finished" status
    if (statusResponse.getEngineStatus() == 3)
    {
        RPEServiceStub.GetOutput getOutput = (RPEServiceStub.GetOutput)
RPEServiceStub.GetOutput.class.newInstance();
        getOutput.setGetOutput(serviceRunResponse);

stub.getOutput(getOutput).getGetOutputResponse();

        URL url = new URL("http://localhost:8080/rpe/services/
RPEService?wsdl");

        String remoteResultPageURL = url.getProtocol() + "://" +
url.getHost() + ":" + url.getPort() + "/rpe/DownloadPage?reportID=" +
String.valueOf(serviceRunResponse);
        System.out.println("URL of the result page: " +
remoteResultPageURL);

        String remoteResult = url.getProtocol() + "://" + url.getHost()
+ ":" + url.getPort() + "/rpe/DownloadFile?reportID=" +
String.valueOf(serviceRunResponse);
        System.out.println("URL of the result archive: " +
remoteResult);
    }
}
```

Command Line Switches for rpe-launcher.exe

"%RPE_HOME%\launcher\rpe-launcher.exe" [-switch] [parameter]

Switch	Notes
	By passing to rpe-launcher.exe only the path to a Document Specification file, it will be open in Launcher.
-wizard	Starts the RPE Launcher minimized, with the 'Publish' wizard started in the foreground. This switch should be followed by an optional path to a definition file, to pre-configure the data source contained in the template selected in the wizard.
-noresult	If this flag is specified in the command line, the 'Results' dialog will not be shown.
-makedocspec	If this flag is provided, along with the path to a definitions file, the RPE Launcher will create a Document Specification file.
-publish	If this flag is provided, along with the path to a Document Specification file, the application will start the publishing process. Can also be used with the -makedocspec parameter to publish the Document Specification created from the definitions file.
-err	If this parameter is missing, non-critical errors that occur during the publishing process are ignored.

Table 4 Valid switches for rpe-launcher.exe

The structure of a sample definitions file is shown below. It consists of a definition file that can be used to configure Tau and DOORS data sources.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<definition xmlns="http://www.ibm.com/rational/rpe/definitionFile"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.ibm.com/rational/rpe/definitionFile
definitionFile.xsd">

    <docspec>c:\temp\docspec\generatedds.dsx</docspec>

    <datasource type="DOORS">
        <property name="URI">/Rational Publishing Engine - RPE/
Requirements/RPE - User Requirements</property>
```

```
<property name="module_id">00000024</property>
<property name="baseline">Current</property>
<property name="view">Object Type View</property>
</datasource>

<datasource id="DOORS 1" type="DOORS">
  <property name="URI">/demo/car/System requirements</
property>
  <property name="baseline">Current</property>
  <property name="view">Industrial</property>
</datasource>

<datasource id="" type="Tau">
  <property name="URI">C:\Program
Files\IBM\Rational\TAU\4.3\examples\umlVerificationCoffeeMachine\CMdesig
n.ttp</property>
</datasource>

<template>C:\Program Files\IBM\Rational\Rational Publishing
Engine\1.1\source\DOORS\examples\doorsData.dta</template>

</definition>
```

The <docspec> section instructs the RPE Launcher to save the generated Document Specification in a certain location.

Each <datasource> section specifies a data source type along with their properties.

Each <template> section specifies a path to the template to be added to the Document Specification.

Appendix: Notices

© Copyright 2000, 2009

U.S. Government Users Restricted Rights - Use, duplication, or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send written license inquiries to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send written inquiries to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions. Therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Intellectual Property Dept. for Rational Software
IBM Corporation
1 Rogers Street
Cambridge, Massachusetts 02142
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Copyright license

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at www.ibm.com/legal/copytrade.html

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product or service names may be trademarks or service marks of others.