

Rational Integration Tester



# Reference Guide for Java Objects

*Version 8.0.0*



**Note**

Before using this information and the product it supports, read the information in “Notices” on page 10.

This edition applies to version 8.0.0 of Rational Integration Tester and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright IBM Corporation 2001, 2012.**

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

# Contents

<b>About this Publication</b> .....	<b>iv</b>
Intended Audience .....	v
Scope .....	v
Typographical Conventions .....	v
Contacting IBM Support .....	v
<b>Testing Java Objects</b> .....	<b>1</b>
Overview .....	2
Adding Java Object Schemas .....	3
Java Objects in Messages .....	4
Example .....	7
<b>Glossary</b> .....	<b>9</b>
<b>Notices</b> .....	<b>10</b>
Trademarks and service marks .....	13

# About this Publication

## **Contents**

### **Intended Audience**

### **Scope**

### **Typographical Conventions**

### **Contacting IBM Support**

This guide describes how to use IBM® Rational® Integration Tester to test Java objects, including serialized POJOs and serialized JavaBeans.

---

## Intended Audience

This document intended to be read by those with a fair understanding and exposure to the concepts involved in both testing and development and in enterprise integration.

## Scope

This document discusses the use of IBM Rational Integration Tester to test Java objects. Information about other features and functionality in Rational Integration Tester is beyond the scope of this document.

If you wish to familiarize yourself with Rational Integration Tester, please refer to the online help or any of the documentation that is provided with the product.

## Typographical Conventions

The following typographical conventions are observed throughout this document:

---

Type	Usage
Constant Width	Program output, listings of code examples, file names, commands, options, configuration file parameters, and literal programming elements in running text.
<i>Italic</i>	Document title names in statements that refer you to other documents. Also used to highlight concepts when first introduced.
<b>Bold</b>	Menu items in graphical user interface windows (such as Microsoft Windows-based or UNIX X Window applications) from which you select options or execute macros and functions.  Submenus and options of a menu item are indicated with a “greater than” sign, such as <b>Menu &gt; Submenu</b> or <b>Menu &gt; Option</b> .

---

## Contacting IBM Support

To contact IBM Support, see: [www.ibm.com/contact/us/en/](http://www.ibm.com/contact/us/en/)

# Testing Java Objects

## **Contents**

### **Overview**

### **Adding Java Object Schemas**

### **Java Objects in Messages**

### **Example**

This chapter provides information about how to test Java objects using Rational Integration Tester.

---

## 1.1 Overview

Java objects (JAR files) can be added to Rational Integration Tester and applied to messages like a schema to test the classes that the JAR file contains by sending Java objects and performing validation against received objects.

The objects to be tested must be part of a JAR file that is added like a schema to Rational Integration Tester's Schema Library in the Architecture School perspective.

Any classes in the JAR file that are public and implement the `java.io.Serializable` interface will be made available as roots in the schema.

**NOTE:** If the JAR file from which you wish to generate the schema has dependencies on other classes, those classes should be present in one or more JAR files co-located with the JAR file from which the schema is being generated. Otherwise, the message formatter may not behave as expected.

**NOTE:** Both bean properties and public fields can be fields within a root. If a bean property and a public field share the same name, the property is used rather than the field.

The classes can then be selected in the normal way from the schema wizard (for example, when applying the schema to a selected message).

**NOTE:** Lists and Sets are supported as fields, and each element can be either primitive type or one of the roots defined in the schema. Arrays and Maps are currently **not** supported.

**NOTE:** A Java object schema can be applied only to byte array fields (for example, messages using the Byte Array formatter) or to messages of type Bytes or Object (set within the message body when using any of the JMS providers).

---

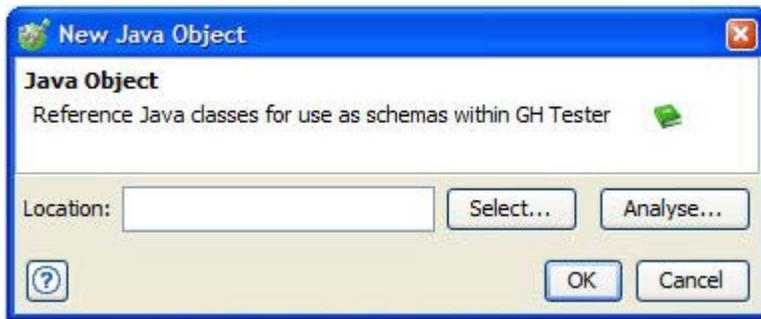
## 1.2 Adding Java Object Schemas

This section describes how to add a Java object schema (JAR file) to the Schema Library in Rational Integration Tester's Architecture School perspective.

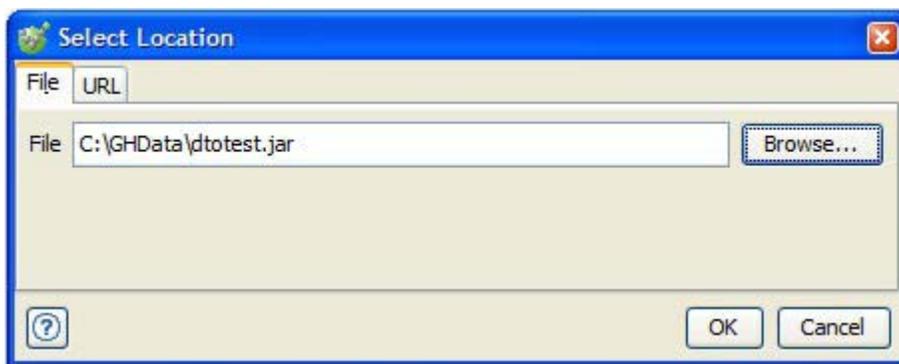
1. Open the Architecture School perspective (**F7**) and select the Schema Library view.
2. Click the Java Object icon in the Schema Library toolbar.



3. In the **New Java Object** dialog, click **Select** to select the JAR file that contains the classes you want to test.



4. In the Select Location dialog, click Browse to locate and select a local JAR file, or click the URL tab to enter the URL of a remote file.



5. Click **OK** when finished, and click **OK** in the New Java Object dialog.
6. The JAR file is now available to be applied to messages as a schema, and the objects and the public member variables they contain can be viewed in the Schema Library.

---

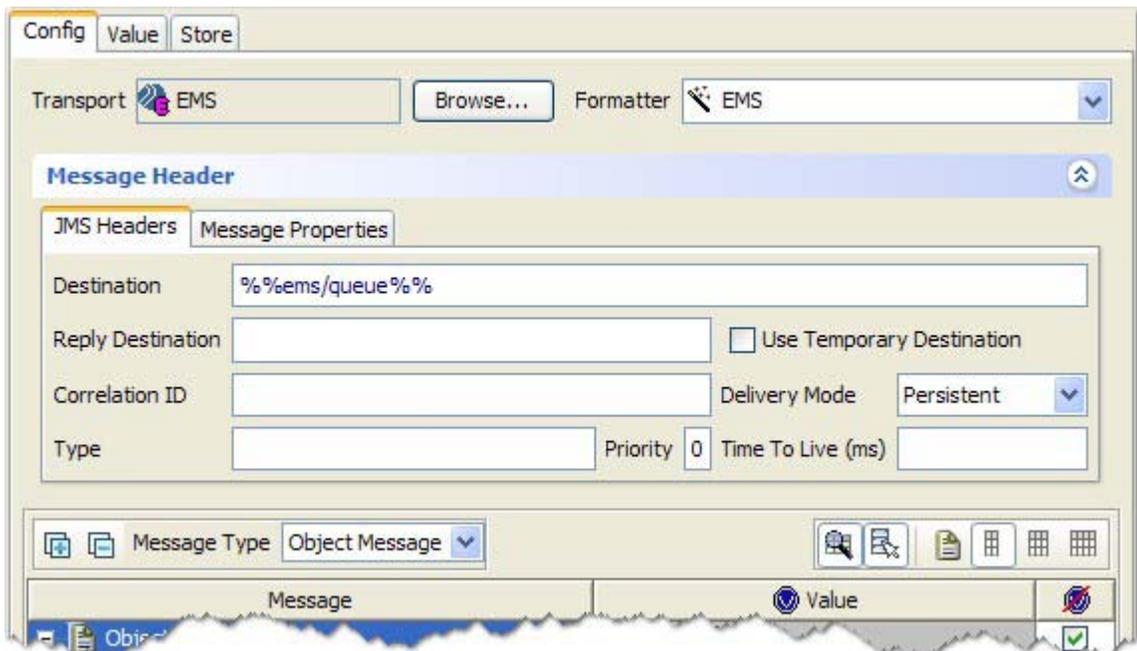
## 1.3 Java Objects in Messages

Once your JAR file has been added to Rational Integration Tester, it can be applied as a schema to messages in applicable test actions. The following example illustrates how to apply an existing Java object schema to a message.

1. Create a new test in an operation that contains a reference to your JMS provider.
2. Add a **Publish** action followed by a **Subscribe** action to the Test Steps phase of the test.



3. Double-click the **Publish** action to open it for editing.
4. Select your JMS-based transport and formatter, and select **Object Message** in the **Message Type** field (above the message body).



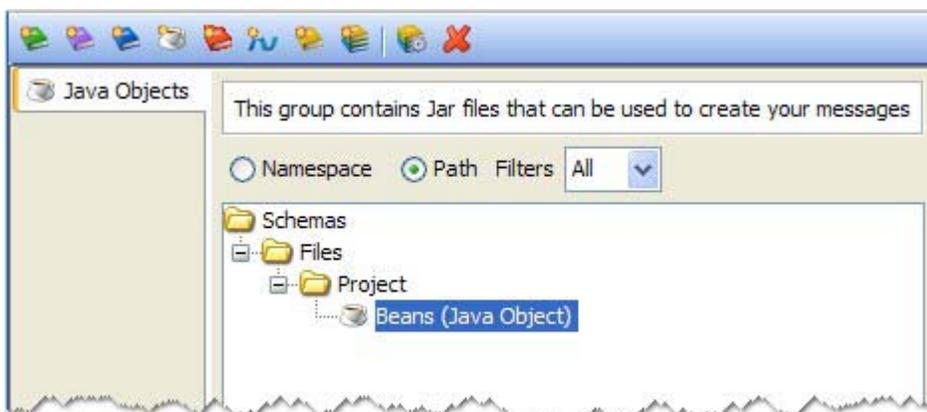
5. In the message area, right-click the **Object (Message)** element and select **Root > Object** from the context menu.



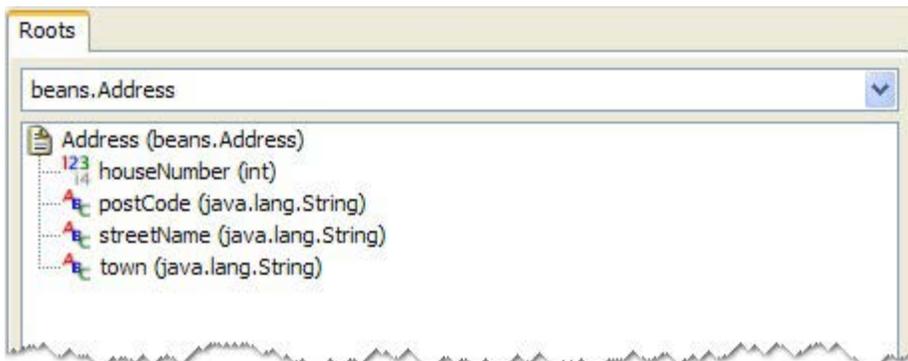
6. When prompted whether or not to retain the content of the existing message body, select **No, overwrite content**.
7. Right-click the **object (pojo)** element and select **Schema...** from the context menu



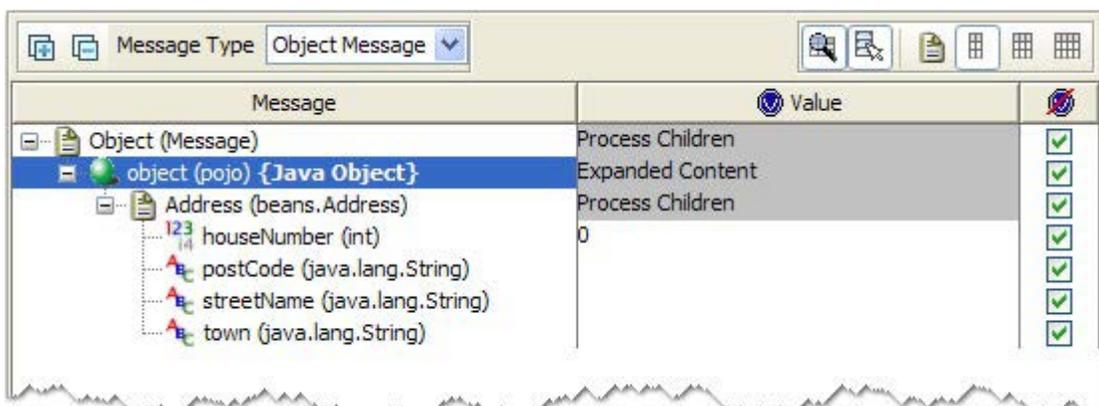
8. In the **Select Schema** wizard, select the schema that contains the object you want to apply to the message (this is in the JAR file that was imported into Rational Integration Tester's Schema Library).



- 
- Under the **Roots** tab on the right side of the wizard, select the desired object.



- Click **Next** to proceed and set the desired Content and Assert options in the next wizard dialog, then click **Finish**.
- The schema will be applied to the original message, and the valid members defined in the JAR file will be message fields.



- The same steps can be carried out in the Subscribe action to apply validation options.

**NOTE:** For more information about messages, schemas, and validation, refer to *IBM Rational Integration Tester Reference Guide*.

---

## 1.4 Example

The following class illustrates how members of a class are translated into fields in the schema. Note that in the resulting schema, “lastName” is a String rather than an int, as the property was used rather than the public field of the same name.

### Code

```
package beans;

public class Name implements java.io.Serializable {

    private static final long serialVersionUID =
2838957914515043754L;

    private String m_firstName;
    private String m_lastName;
    public int lastName; // This will not appear in the schema.
    public String middleName; // This will appear in the schema.

    public String getFirstName() {
        return m_firstName;
    }

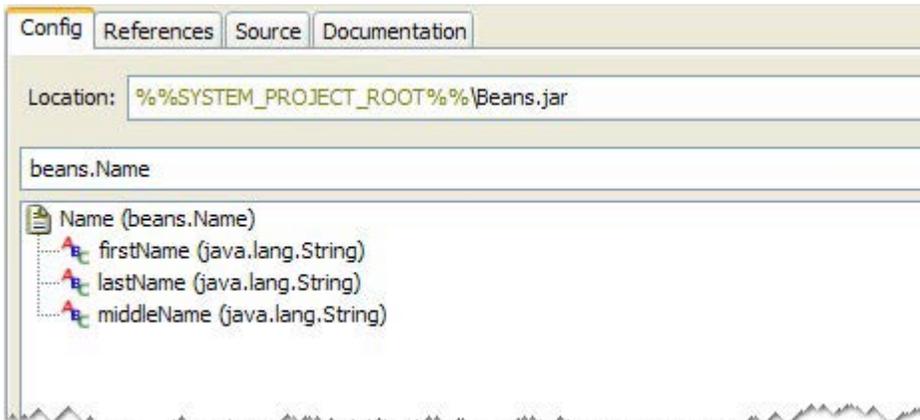
    public void setFirstName(String firstName) {
        m_firstName = firstName;
    }

    public String getLastName() {
        return m_lastName;
    }

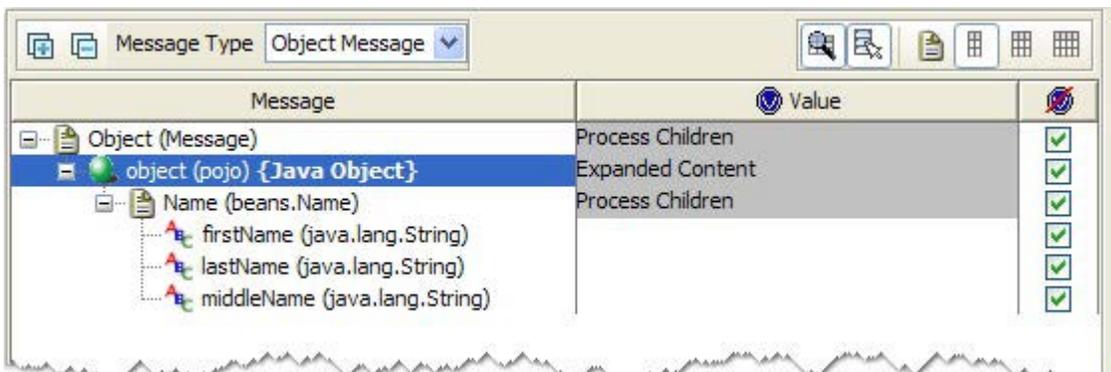
    public void setLastName(String lastName) {
        m_lastName = lastName;
    }
}
```

---

## Schema Library



## Message



---

# Glossary

The following table below lists some of the key terms used in this document, and provides a description of each.

---

<b>Term</b>	<b>Description</b>
Field	A bit of data constituent to a message. Most fields are scalar and therefore unitary, equivalent to data attributes. Vector fields are an aggregation of fields both scalar and vector, and are usually referred to as Messages. See also Message.
Message	A unit of information made up of a header consisting of meta-information and a body consisting of the message data.
Host	The computer on which a software process runs.
Publisher-Subscriber	A messaging paradigm whereby a messaging network consists of Publishers and Subscribers.
Transport	Informally, the messaging software in use. For instance, TIBCO Rendezvous, TIBCO ActiveEnterprise, IBM WebSphere® MQ (JMS).
Publishing	Making a message (data) available on a message channel.
Subscribing	Receiving a stream of messages (data) on a given message channel.
Server	A host computer on a network shared by more than one user.

---

---

# Notices

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 license inquiries, in writing, to:

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

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

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
1623-14, Shimotsuruma, Yamato-shi  
Kanagawa 242-8502 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.

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:

IBM United Kingdom Limited  
Intellectual Property Law  
Hursley Park  
Winchester  
SO21 2JN  
Hampshire  
United Kingdom

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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.

#### 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 Corporation 2001, 2012.

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

---

## Trademarks and service marks

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 “Copyright and trademark information” at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

**IBM**<sup>®</sup>