

**Oracle Insurance**

**Insbridge Rating and  
Underwriting  
SoftRater User Guide**

Release 4.6

May 2012

Copyright © 2005, 2012, Oracle and/or its affiliates. All rights reserved.

Oracle Insurance Insbridge Rating and Underwriting SoftRater User Guide

Release 4.6

Part # E35541-01

Library # E35552-01

May 2012

Primary Author: Mary Elizabeth Wiger

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

#### **U.S. GOVERNMENT RIGHTS**

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Where an Oracle offering includes third party content or software, we may be required to include related notices. For information on third party notices and the software and related documentation in connection with which they need to be included, please contact the attorney from the Development and Strategic Initiatives Legal Group that supports the development team for the Oracle offering. Contact information can be found on the Attorney Contact Chart.

The information contained in this document is for informational sharing purposes only and should be considered in your capacity as a customer advisory board member or pursuant to your beta trial agreement only. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described in this document remains at the sole discretion of Oracle.

This document in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle Software License and Service Agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced, or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

# CONTENTS

	PREFACE.....	V
	Audience.....	v
	Related Documents.....	v
	Conventions.....	vi
	System Requirements.....	vi
	Manual History.....	vi
<b>CHAPTER 1</b>		
	INTRODUCTION TO SOFTRATER.....	7
<b>CHAPTER 2</b>		
	INTRODUCTION TO SOFTRATER CACHING.....	9
	What is a Cache Hit? .....	9
	What is a Cache Miss?.....	10
	When is Information Removed from the Cache?.....	11
	Cache Use Exception.....	11
<b>CHAPTER 3</b>		
	SOFTWARE INTEGRATION FOR WINDOWS.....	12
	Web Services Interfaces for Windows .....	12
	Using HTTP .....	12
	SOAP Example HTTP SOAP Proxy .....	13
	Examples.....	14
	JavaScript (Web Services) .....	14
	C# Example (Web Services) – Process Message .....	14
	C# Example (Web Services) – Custom Message.....	15
<b>CHAPTER 4</b>		
	RATING ARGUMENTS FOR WINDOWS.....	16
<b>CHAPTER 5</b>		
	CUSTOM XML ARGUMENTS FOR WINDOWS.....	17
<b>CHAPTER 6</b>		
	INSBRIDGE.XML WINDOWS EXAMPLE .....	19
	Insbridge.XML Request Format.....	19
	Insbridge.XML Result Format.....	22
	Examples.....	27

<b>CHAPTER 7</b>	
	SOFTWARE INTEGRATION FOR JAVA..... 30
	Software Integration Methods..... 30
<b>CHAPTER 8</b>	
	RATING ARGUMENTS FOR JAVA..... 32
<b>CHAPTER 9</b>	
	CUSTOM XML ARGUMENTS FOR JAVA..... 33
<b>CHAPTER 10</b>	
	INSBRIDGE.XML JAVA EXAMPLE..... 34
	Insbridge.XML Request Format..... 34
	Insbridge.XML Result Format..... 38
	Examples..... 42
<b>NOTICE</b>	
	UPGRADE NOTICE..... 47
<b>SUPPORT</b>	
	CONTACTING SUPPORT ..... 48
	TTY Access to Oracle Support Services..... 48
<b>INDEX</b>	
	INDEX ..... 49

## LIST OF FIGURES

FIGURE 1 RESULT CODE WINDOWS .....	21
FIGURE 2 SOFTRATER SWITCHES .....	37

## PREFACE

Welcome to the *Oracle Insurance Insbridge Rating and Underwriting SoftRater User Guide*. This guide describes the concepts and usage of Oracle Insurance Insbridge Rating and Underwriting SoftRater (SoftRater). This guide describes the concepts and requirements of SoftRater. It provides a reference for developers to properly interact with the Insbridge SoftRater Engine either through SOAP, POST Web Services Interface (WSI) or Direct EJB Interfacing.

This guide contains reference information on these SoftRater engines:

- SoftRater for Windows
- SoftRater for WebSphere
- SoftRater for WebLogic
- SoftRater for JBoss

### Document Composition

Individual reference guides for SoftRater for WebSphere, SoftRater for WebLogic and SoftRater for JBoss are now consolidated under the SoftRater for Java title. SoftRater for Java provides one central location to reference materials that apply to all Java platforms. Installation instructions are available for each engine platform.

SoftRater for Windows references will be included in the SoftRater User Guide as well.

## AUDIENCE

This guide is intended for SoftRater system administrators who are tasked with administering SoftRater. Readers of this guide should be familiar with XML, HTTP.

## RELATED DOCUMENTS

For more information, refer to the following Oracle resources:

- The Oracle Insurance Insbridge Rating and Underwriting Framework Administrator User Guide.
- The Oracle Insurance Insbridge Rating and Underwriting SoftRater Server User Guide.
- You can view these guides in-line at this address:

<http://www.oracle.com/technetwork/documentation/insurance-097481.html>

## CONVENTIONS

The following text conventions are used in this document:

Convention	Description
<b>bold</b>	Boldface type indicates graphical user interface elements associated with an action.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

## SYSTEM REQUIREMENTS

For minimum operating system and hardware requirements, please see the Hardware Software requirements guide.

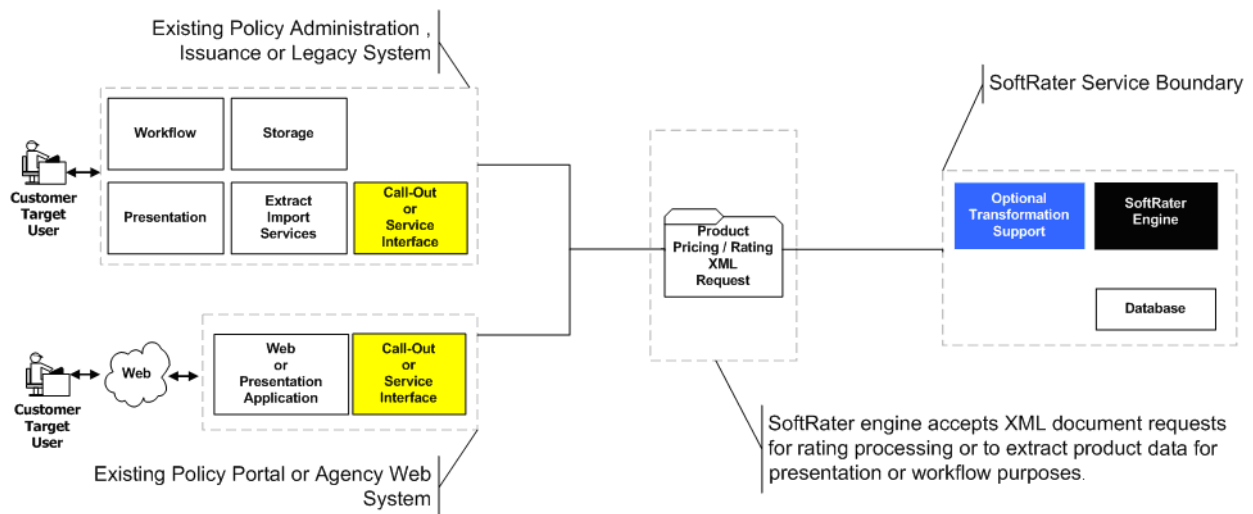
## Manual History

New editions incorporate any updates issued since the previous edition.

Edition	Publication Number	Product Version	Publication Date	Comment
15 <sup>th</sup> Edition	P01-721-15	R 4.0	April 2010	Update Version and Combine
16 <sup>th</sup> Edition	P01-721-16	R 4.0.1	August 2010	Update Release
17 <sup>th</sup> Edition	P01-721-17	R 4.1	December 2010	Update Release
18 <sup>th</sup> Edition	P01-721-18	R 4.5	May 2011	Update Release
19 <sup>th</sup> Edition	P01-721-19	R 4.5.1	September 2011	Update Release
20 <sup>th</sup> Edition	P01-721-20	R 4.6	May 2012	Update Release

## INTRODUCTION TO SOFTRATER

SoftRater is the rating engine portion of the Oracle Insurance Insbridge Rating and Underwriting (IBRU) system. The SoftRater rating engine is updated with logic created in RateManager and is the run-time environment for a company's rates. SoftRater supports request integration through a Web Services (SOAP) interface or through native Java or .NET interfaces. The engine is multi-platform and has the ability to process rates in both a transactional and batch mode.



The following application sources are supported by Oracle Insurance Insbridge Rating and Underwriting:

- Web applications
- Server based applications
- Mainframe applications
- Or any combination of the above.

The SoftRater Engine performs all calculations and underwriting and outputs Insbridge XML, which is reformatted for distribution to all application sources.

**For Windows:** SoftRater utilizes Microsoft SQL Server 2005 SP/2 and 2008 SP/1 as its data store.

**For Java:** SoftRater can utilize Microsoft SQL Server 2005 SP/2, and 2008 SP/1, Oracle 10g and 11g, and IBM DB2 v9.5.

The platforms for SoftRater engines are:

- SoftRater for Windows
- SoftRater for Java, consisting of:
  - SoftRater for WebSphere
  - SoftRater for WebLogic
  - SoftRater for JBoss



## SoftRater for Windows

- SoftRater for Windows is tested and certified on Microsoft Windows Server 2003 and Windows Server 2008.

## SoftRater for Java

- SoftRater for JBoss is tested and certified on Red Hat JBoss Application Server Versions 4.2.3.

**Version:** 4.2.3.GA (build: SVNTag=JBoss\_4\_2\_3\_GA date=200807181418)

**Version Name:** Trinity

**Built on:** July 18 2008

- SoftRater for WebSphere is tested and certified on IBM WebSphere V 6.1.
- SoftRater for WebLogic is tested and certified on Oracle WebLogic Release 11gR1 (10.3.1)
  - Premium edition
  - Advantage edition
  - Workgroup edition

---

**NOTE:** *WebLogic Express (WLX) edition is not currently supported.*

---

---

**NOTE:** *JBoss, WebLogic and WebSphere operate in a variety of environments. As long as the application server is compatible with the operating system, then SoftRater for Java also will be compatible. SoftRater for Java is installed and functions within application server. The external resources that the application server resides in will not affect SoftRater for Java.*

---

## Guide Update

Individual reference guides for SoftRater for WebSphere, SoftRater for WebLogic and SoftRater for JBoss are now consolidated under the SoftRater for Java title. SoftRater for Java provides one central location to reference materials that apply to all Java platforms. Chapters in this guide that pertain to SoftRater for Java will be identified.

SoftRater for Windows reference guide will also be added to the SoftRater User Guide. Chapters in this guide that pertain to SoftRater for Windows will be identified.

# INTRODUCTION TO SOFTRATER CACHING

The SoftRater engine enables SoftRater to store mapped variable information in memory for quick access. This reduces the amount of information that must be pulled from the database and dramatically improves rating time.

The cache is broken down by line of business, program, version and environment. This means that information for the following could all exist in the cache at the same time, independent of one another (i.e. removing or updating one, does not remove or update any of the others):

LOB	Program	Version	Environment
1	1	1	Dev
1	1	1	Prod
1	1	2	Dev
1	2	1	Dev
2	1	1	Dev

For more information, see the following topics:

- What is a Cache Hit?
- What is a Cache Miss?
- When is Information Removed from the Cache?
- Cache Use Exceptions

For information on configuring the cache, see the Insbridge Framework Administrator topic Introduction to SoftRater Engine Configuration and the SoftRater Server topic Program Cache.

## WHAT IS A CACHE HIT?

Any time a value is found in a mapped variable table, for a specific set of criteria, it is considered a cache hit. All hits are stored in the cache. For example, consider the following mapped variable table:

Variable	Criteria			
Territory Code	ZIP Code	City	County	State
1	75080	Richardson	Dallas	Texas
2	75081	Richardson	Dallas	Texas
3	75082	Richardson	Dallas	Texas
4	75083	Richardson	Dallas	Texas
5	75085	Richardson	Dallas	Texas

Then the following data request would be considered a hit and a value of 3 would be stored for the mapped variable:

**ZIP Code:** 75082  
**City:** Richardson  
**County:** Dallas  
**State:** Texas

If this same request had not been made previously, then a call would be made to the database to retrieve the information. Once obtained, the information would be stored in the cache as a hit. A subsequent request with this same information would result in the information being found in the cache, and a call to the database would not be necessary.

## WHAT IS A CACHE MISS?

Anytime a value is not found in a mapped variable table, for a specific set of criteria, it is considered a cache miss. All misses are stored in the cache however; the Category Miss Buffer Purge % can be set to purge a certain percentage of the misses. Consider the following mapped variable table:

<b>Variable</b>	<b>Criteria</b>			
<b>Territory Code</b>	<b>ZIP Code</b>	<b>City</b>	<b>County</b>	<b>State</b>
1	75080	Richardson	Dallas	Texas
2	75081	Richardson	Dallas	Texas
3	75082	Richardson	Dallas	Texas
4	75083	Richardson	Dallas	Texas
5	75085	Richardson	Dallas	Texas

Then the following data request would be considered a miss and the default value for the mapped variable would be used:

**ZIP Code:** 75084  
**City:** Richardson  
**County:** Dallas  
**State:** Texas

If this same request had not been made previously, then a call would be made to the database to retrieve the information. When a value is not found in the database for the request, the default value would be used and the information would be stored in the cache as a miss. A subsequent request with this same information would result in the information being found in the cache, and a call to the database would not be necessary (unless the miss had been purged from the database).

## WHEN IS INFORMATION REMOVED FROM THE CACHE?

Information is removed from the cache only in the following cases:

- Caching is disabled on the SoftRater Engine Configuration screen in the Insbridge Framework Administrator, and the change is saved by clicking **SAVE**. In this case, all information is removed from the cache for all programs.
- A new package is loaded to an environment for a program that is currently cached. In this case, all information is removed only for that specific program and that specific environment.
- An update is made to a subscriber's SoftRater Explorer configuration settings (i.e. an environment is added, deleted or edited; a virtual file server is added; etc.). In this case, all information is removed from the cache for all programs.
- The server is restarted or shutdown.

---

**NOTE:** *Cache will equal true even when the cached program is manually removed from IBFA. As long as cache is on, the Insbridge Engine will use the cache that the current rate request enters itself or from other requests.*

---

## CACHE USE EXCEPTION

There is an occasion when SoftRater will not use the cache when rating.

- **The environment being rated against has a Catalog Type of RateManager.** The RateManager environment is the location that all local packages are loaded to upon creation. Typically, the RateManager environment is only rated against during program development or to debug a rating issue. This is usually done through Testing; however, it can be done through the SoftRater Test Interface as well. Since the development environment is very dynamic, caching would actually slow down rating in most cases.

# SOFTWARE INTEGRATION FOR WINDOWS

The Web Service Interface (WSI) provides a platform, environment, and language neutral mechanism for business process interoperability using two common denominators of the Internet, XML and HTTP. The SoftRater WSI is supported through the use of Microsoft Internet Information Server (IIS) and ASP.NET.

Supported operations are SOAP, HTTP POST, and HTTP GET. In order to utilize the SoftRater rating arguments, the document must be a SOAP envelope.

Upon request, the URL configures an instance of the SoftRater Engine from those arguments and forwards the InsbridgeRate.XML (Input format) to the SoftRater Engine for processing. After the SoftRater Engine completes processing, the result, InsbridgeRate.XML (Output format), is forwarded through (HTTP) back to the requesting process.

When rating custom XML, the engine has the option of stateful rating. The SoftRater WSI will add the contents of the rating results to the document that was submitted for rating. This is an important consideration when constructing XSLT (mapping) files.

---

**NOTE:** If you are submitting Insbridge XML using a testing tool, it is recommended that you use HTTP POST instead of SOAP. The SOAP should only be used when rating custom XML. If present or if you are using .NET SOAP remove <MappedRateRequest> section from the SOAP header.

---

## WEB SERVICES INTERFACES FOR WINDOWS

SoftRater provides Web Service Interface for a SoftRater Rating XEngine. The following operations are supported. For a formal definition, please review the Service Description found on your IBFA instance at <http://<yourserver>/ibfa/connectors/sofrater.asmx?>.

- **Ping** – This will obtain the status of the Insbridge XEngine.
- **ProcessAsyncMessage** – Submit Rate Request to the Insbridge XEngine Rate Broker.
- **ProcessCustomMessage** – Obtain Rates from the Insbridge XEngine using custom xml. You also can use this method for an Insbridge.XML request. The Java returned is in string format and is not loaded in a system.XML document object. ProcessCustomMessage should be used for non-.NET communication, (i.e. Java to Java or Java to .NET). .NET to .NET communication also can use ProcessCustomMessage.
- **ProcessMessage** – Obtain Rates from the Insbridge XEngine. Custom XML is supported. The return file is loaded into a system.XML document object. ProcessMessage should be used for .NET to .NET communication only.
- **ReceiveAsyncMessage** – Obtain Rate Results from the Insbridge XEngine Rate Broker.

## Using HTTP

There are two HTTP Web Services:

1. **HTTP SOAP Proxy** – SoftRater Web Service – From the WSDLs, proxy classes can be generated in a SOAP supported development environment that communicate with the installed SoftRater instance. The SoftRater Web Service WSDL is located at the following URL.  
<http://yourserver/ibfa/connectors/sofrater.asmx?WSDL>.
2. **HTTP POST** – A lite-weight Web Service Interface. An ASPX page is provided as an interface for clients with only web form POST abilities without using a SOAP envelope message. CustomXml cannot be submitted from this interface. The URL to the POST interface is:  
<http://yourserver/ibfa/Connectors/sofrater.asmx/ProcessCustomMessage>

## Using External Testing Tools

If you are submitting the SOAP request using an external third party testing tool, such as eviware soapUI, you may need to wrap your rating request in a CDATA wrapper.

Please refer to the WSDL for the SOAP parameter data types and valid values.

## Using CDATA

IBDOC and CDATA are containers for the input XML for .NET SOAP. For Java SOAP no IBDOC is used, only CDATA is used as follows:

```
<XMLInputs xsi:type="xsd:string">
<![CDATA[
custom XML....
]]>
</XMLInputs>
```

---

**NOTE:** *If you are submitting Insbridge XML, you should use HTTP POST instead of SOAP. SOAP should be used only when submitting custom XML.*

---

## SOAP Example HTTP SOAP Proxy

When using a testing tool, choose <http://<yourserver>/ibfa/connectors/sofrater.asmx?> and add WSDL to the end of the URL: <http://<yourserver>/ibfa/connectors/sofrater.asmx?WSDL>.

## Examples

### JavaScript (Web Services)

```
var MyResults;
var MyXMLDoc;
var AppURL;
var objHTTP;

MyXMLDoc = "ibDoc=<ibdoc><rate>..... </rate></ibdoc>"; // Well formed InsbridgeRate.XML (Input
// or CustomXml
AppURL = "http://yourserver/ibfa/Connectors/sofrater.aspx/ProcessCustomMessage" // Web Services Request URL
objHTTP = new ActiveXObject("Msxml2.XMLHTTP "); // Microsoft HTTP Request Object

objHTTP.open('post', AppURL, false);

objHTTP.setRequestHeader("SOAPAction","http://insbridge.net/wsi/Connector/SoftRater/ProcessCustomMessage"); //Set the SOAP action
objHTTP.setRequestHeader("Content-Type","application/x-www-form-urlencoded");

objHTTP.send(MyXMLDoc);
MyResults=objHTTP.responseXML; // returns XML response
MyResults=objHTTP.responseText; // returns text response (Optional)
```

### C# Example (Web Services) – Process Message

```
string rateXml = File.ReadAllText("C:\\Insbridge.xml");

//Rate an Instance
pmSoftRater.SoftRater ibSoftRater = new pmSoftRater.SoftRater();

//Rate Operators
pmSoftRater.MappedRateOperators rateOper = new
pmSoftRater.MappedRateOperators();

rateOper.AddHeading = 1;
rateOper.AddRoot = 1;
rateOper.AddInputs = 1;
rateOper.AddResultDesc = 0;
rateOper.AddResultEmpty = 0;
rateOper.DebugRate = 0;
rateOper.EnvRef = "rm_default";

ibSoftRater.MappedRateOperatorsValue = rateOper;

// Rate using the ProcessMessage Service
string results = ibSoftRater.ProcessMessage(rateXml).OuterXml;
```

Created using Visual Studio .NET 2008.

## C# Example (Web Services) – Custom Message

```
string rateXml = File.ReadAllText("C:\\Custom.xml");

//Rate an Instance
pcmSoftRater.SoftRater ibSoftRater = new pcmSoftRater.SoftRater();

//Rate Operators
pcmSoftRater.MappedRateOperators rateOper = new
pcmSoftRater.MappedRateOperators();

//Only for Input and/or Output Transformations
pcmSoftRater.MappedRateRequest rateRequest = new
SRTester.pcmSoftRater.MappedRateRequest();

rateOper.AddHeading = 1;
rateOper.AddRoot = 1;
rateOper.AddInputs = 1;
rateOper.AddResultDesc = 0;
rateOper.AddResultEmpty = 0;
rateOper.DebugRate = 0;
rateOper.EnvRef = "SR";

rateRequest.Subscriber = 1002;
rateRequest.Lob = 1;
rateRequest.Program = 106;
rateRequest.Version = 3;
rateRequest.OutputMappingStateful = false;

// Setting the input/output Transformation templates
rateRequest.InputMappingIdentifier = "customInputXslt.xslt";
rateRequest.OutputMappingIdentifier = "customOutputXslt.xslt";
rateRequest.InputMappingType = pcmSoftRater.MappingType.CUSTOM;
rateRequest.OutputMappingType = pcmSoftRater.MappingType.CUSTOM;

string results = ibSoftRater.ProcessCustomMessage(rateXml, rateOper,
rateRequest);
```

Created using Visual Studio .NET 2008.



## RATING ARGUMENTS FOR WINDOWS

Valid values for rating arguments are entered in these two operations:

### ProcessCustomMessage

```
<ibRateOper>  
... rating arguments  
</ibRateOper>
```

### ProcessMessage

```
<MappedRateOperators>  
... rating arguments  
</MappedRateOperators>
```

The SoftRater engine rating arguments control the handling of XML data out of the system. Rating arguments are optional. For optimal performance, use arguments for your rating integration.

Values: 0= False, 1=True.

- **Add Root Node (Use default – 0)** – If submitting multiple rate request documents, this option is typically set to true to make the result document a valid XML document.
- **Add Inputs (Use default – 0)** – When set to true, the full request Insbridge.XML document is returned in the result Insbridge.XML document making the XML document much larger than normal.
- **Add Heading (Use default – 0)** – When set to true, the program name description information is returned in the result XML also.
- **Add Result Descriptions (Use default – 0)** – When set to true each result item includes the RateManager variable result name along with the result id and value. Making the result XML much larger.

Typically, most integration operates on the result IDs and descriptions are not needed when building an automated system.

- **Add Empty Results (Use default – 0)** – When set to true, a defined result item, whose value is empty (i.e. blank), is still created and returned blank in the resulting Insbridge.XML. If your program design requires a number of optional results, you could have blank results items in your XML.
- **Debug Output (Use default – 0)** – When set to false, no debug report will be issued. Set to true if you would like a debug report.
- **EnvRef (Use default – blank)** – When left blank, the default environment defined in setup will be used. To specifically define an environment, enter in the environment name. Entering a value here will override any value entered in the file.

If no arguments are sent, the default values will be used.

## CUSTOM XML ARGUMENTS FOR WINDOWS

The SoftRater WSI controls the processing (transformations) of XML data in and/or out of the system. Custom XML arguments are required only when you are submitting custom XML. If you are using Insbridge XML, custom XML arguments are not required. The information will be contained in the Insbridge XML. If you are using custom XML and do not define the custom XML arguments, any error message will be thrown.

Valid values for custom XML arguments are entered in these two operations:

### **ProcessCustomMessage**

```
<ibCustomOper>  
... rating arguments  
</ibCustomOper>
```

### **ProcessMessage**

```
<MappedRateRequest>  
... rating arguments  
</MappedRateRequest>
```

Use the following options below for your custom rating integration.

- Lob – Identifier of the Line of Business
- Subscriber – Identifier of the Subscriber
- Program – Identifier of the Program
- Version – Identifier of the Program Version
- InputMappingIdentifier – Name of the Custom Mapping Document
- InputMappingType – Enum for the Custom Mapping Document
  - NONE – No input mapping should be performed
  - GLOBAL – Input mapping is global to the Line of Business. Mapping name required.
  - LOCAL – Input mapping is unique to the program version
  - CUSTOM – Input mapping of the customer that has been added into the workflow. Mapping name required.
- OutputMappingIdentifier – Name of the Custom Mapping Document
- OutputMappingType – Enum for the Custom Mapping Document
  - NONE – No output mapping should be performed
  - GLOBAL – Output mapping is global to the Line of Business
  - LOCAL – Output mapping is unique to the program version
  - CUSTOM – Output mapping of the customer that has been added into the workflow
- OutputMappingStateful – The SoftRater WSI will add the contents of the rating results to the document that was submitted for rating.

- OutputErrorXPathLoc – Location of any system errors that occurred during the Webservice request that are not related to SoftRater for Windows. (By default, an error node is created at the root level.)

---

**NOTE:** *If you are using custom XML to rate or test, the mapping name may need to be passed through. The Input Mapping Type arguments **Global** and **Custom** require the name of the mapping file.*

---

---

**NOTE:** *The **OutputSchema** web services argument is no longer being used. This argument displayed the path of any schema that the WSI should validate against. If you are currently using this, you can leave it in the custom XML.*

---

# INSBRIDGE.XML WINDOWS EXAMPLE

XML is the primary data exchange mechanism used by Oracle Insurance Insbridge Rating and Underwriting system to communicate information electronically with external and internal software systems.

Insbridge's rating request input XML is designed to be flexible and efficient. It allows for single or multiple rate requests to be submitted via one input XML document. The rate requests embedded in this single document can be targeted to multiple states and/or multiple lines of business. Multiple versions of a rating package also can be targeted in a single rate request document.

The rating request response XML is also streamlined to present all the results to the various request methods, described above, in a single output XML document.

## Insbridge.XML Request Format

The following is an example of an Insbridge rate request XML document:

```
<rate lob="2" tracking_attribute="" env_def="">
  <heading>
    <program parent_id="700" program_id="24" program_ver="1"/>
  </heading>
  <c i="0" desc="Policy">
    <m i="1086" n="PackageDisclnd" v=""/>
    <m i="1094" n="RenewalRetentionCreditInd" v=""/>
    <m i="1157" n="CompanyCode" v=""/>
    <m i="1212" n="Eff_Date" v=""/>
    <m i="1214" n="PrimInsuredAge" v=""/>
    <m i="1215" n="SecInsuredAge" v=""/>
    <m i="1222" n="RenewalInd" v=""/>
  <c i="5" desc="Home">
    <m i="1083" n="TerritoryCode" v=""/>
    <m i="1084" n="ResidenceType" v=""/>
    <m i="1087" n="ProtectionClass" v=""/>
    <m i="1095" n="Wood/Tile/SlateRoofType" v=""/>
    <m i="1096" n="HomeDeductible" v=""/>
    <m i="1098" n="WindstormOrHailDeductible" v=""/>
    <m i="1100" n="CentralStationFireAlarmInd" v=""/>
    <m i="1101" n="CentralStationBurglarAlarmInd" v=""/>
  <c i="8" desc="Coverage">
    <m i="1204" n="CovCd" v=""/>
    <m i="1205" n="CovLimit" v=""/>
    <m i="1207" n="CovEff_Date" v=""/>
  </c>
  <c i="9" desc="Endorsement">
    <m i="1181" n="EndorCd" v=""/>
    <m i="1182" n="EndorRateInd" v=""/>
    <m i="1190" n="EndorEff_Date" v=""/>
    <m i="1191" n="Parm5" v=""/>
  </c>
</rate>
```

```
    </c>
  </c>
</rate>
```

### <rate> Node

The <rate> node marks the beginning of a rate request for a specific line of business. This node has the required attribute, “lob”, which identifies the “line of business” for the request. In the following example, the lob attribute is set to “2” which represents “Home” insurance according to the Insbridge standard line of business mappings (see RateManager Admin User Guide). The rate node attributes are defined as follows:

**lob** – line of business indicator per Insbridge standard lob codes (Required)

**env\_def** – rating environment indicator per Insbridge Framework Administrator (Optional). Allows the user to override the default rating environment. By default, SoftRater rates against the default environment, as set up in the Insbridge Framework Administrator (see Introduction to Environments in the Insbridge Framework Administrator User Guide). To rate against a different environment, add the attribute env\_def="Env\_Name" to the rate node, where Env\_Name is the name of the environment you wish to rate against.

Example:

```
<rate lob="1" env_def="Env_Name">
```

**renc** – Allows the user to instruct SoftRater to not encode XML characters that are not considered valid XML characters. These characters are:

As an optional feature, all other attributes provided on the <rate> node are collected as tracking attributes to be returned in the result XML document as attributes in the <result> node. This allows the original rate request to be uniquely tracked with its result XML document by any identification elements available to the calling subsystem. In the example below, the “ policyId='A1206' ” attribute value pair would be mirrored on the <result> node of the resulting output XML.

Example:

```
<rate lob="2" policyId="A1206">
```

This rate request may be targeted to one or more rating logic instances based on what is found in the <heading> node (described) next.

### <heading> Node

The <heading> node serves only as a container for <program> nodes and has no attributes. If multiple <program> nodes are found in the heading node, then rating is performed for each node, if possible, and appropriate results are generated in the output XML.

Example:

```
<heading>
  <program parent_id="700" program_id="24"
    program_ver="1"/>
  <program parent_id="700" program_id="22"
    program_ver="1"/>
</heading>
```

### <program> Node

The <program> node specifies a specific SoftRater Package (rating engine logic instance) to run this rate request against. A program typically represents rating logic for a particular State and line of business (e.g.: Texas Auto insurance, California Home insurance). The program node attributes are defined as follows:

- parent\_id - Insbridge identifier typically assigned to a carrier for database lookup purposes (Required)
- program\_id - Insbridge identifier assigned to a program (rating engine logic instance) which represent the rating rules necessary to generate a quote. (Optional)
- program\_ver - A particular version of a program. Each version may have different rating rules, inputs, outputs, etc. (Optional)

#### Attribute Requirement Rules:

1. If the `program_id` is provided then the version `program_ver` value is used to select a program version to be processed against the source XML.  
Example: `<program parent_id="100" program_id="100" ver="3"/>`
2. If the `program_id` is provided and the `program_ver` is not, the version selection rules for the program, identified by `program_id`, are used to select the program version to be processed against the source XML.  
Example: `<program parent_id="100" program_id="100" />`

The <program> node allows mapped input overrides to be specified for a specific program in a multi-program (or multi-state) rate request. See Multiple Rate Requests in a Single XML Document for more information.

**result\_def** – Allows the user to override the default result mapping group. By default, SoftRater uses the group that is set up as the default in RateManager. To use a different result group, add the attribute `result_def="Result_Code"` to the rate node, where `Result_Code` is the result code shown in RateManager.

```
<program parent_id="123" program_id="1" program_ver="1" result_def="CA7E730716"/>
```

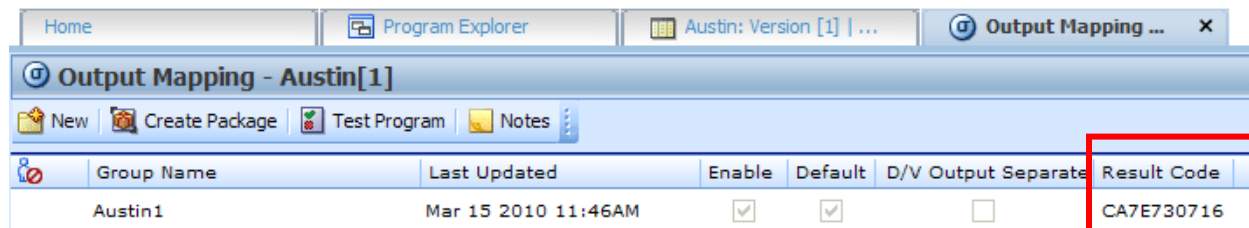


Figure 1 Result Code Windows

In order for you to be able to use a result group, it must have been **Enabled** when the package was created (see the RateManager topic Editing Result Group Options).

### <c> Node (Category)

The <c> (category) node marks the beginning of input data for a specific category of information (i.e.: home, car, driver, policy, etc). It has the following attributes:

- i - identifier. Zero always indicates “Policy” level inputs, other categories are user definable.  
(Required)
- desc - description of category (Optional)

Example: <c i="0" desc="Policy">

The “Policy” category is always a direct child of the <rate> node, except for overrides in the <program> node (described later). It is also the top-level category node. The Policy category node typically has an ID of zero (i.e.: i="0"). Item level categories are nested under the Policy category node.

Example:

```
<c i="0" desc="Policy">
  <m i="1212" n="Eff_Date" v="" />
  <m i="1214" n="PrimInsuredAge" v="" />
  <m i="1215" n="SecInsuredAge" v="" />
  <c i="5" desc="Home">
    ....
    <c i="7" desc="Scheduled Property">
      ....
    </c>
  </c>
  <c i="5" desc="Home">
    ....
  </c>
</c>
```

### <m> Node (Map)

The <m> (map) node represents an individual attribute-value pair mapped for a specific SoftRater Package hosted by SoftRater. In the input case, it identifies an input attribute recognized by the SoftRater Package (or Packages) listed in the <heading> node, and its associated value. The <m> node’s attributes are:

- i - input identifier (Required)
- n - name, description of input (Optional)
- v - value of input (Required)

The <m> node is always a child of a <c> (category) node and is an attribute-value pair for that specific category instance (see the Policy category example).

## Insbridge.XML Result Format

The following is an example of an Insbridge rate request result XML document. The result XML is very similar to the input XML format.

```
<result lob="2" st="6" gen_date="2/19/2010 1:55:32 PM">
  <program parent_id="900" program_id="1" program_ver="1" status="PASS">
    <c i="5">
      <m i="Dwelling_1" v="640"/>
      <m i="Dwelling_3" v="0"/>
    </c>
  </program>
</result>
```

```

    <m i="Dwelling_5" v="0"/>
    <m i="replcc" v="0"/>
    <m i="COV Replacement Cost Contents" v="0"/>
    <m i="COV Mortgage Payment" v="0"/>
    <m i="Dwelling_10" v="0"/>
    <m i="Dwelling_11" v="34"/>
    <m i="COV Replacement Cost - Dwelling" v="0"/>
    <m i="Dwelling_13" v="0"/>
    <m i="Dwelling_13" v="674"/>
    <m i="COV Business Pursuits" v="0"/>
    <m i="COV Permitted Incidental Occupancies" v="0"/>
    <m i="COV Personal Injury" v="0"/>
    <m i="Dwelling_17" v="0.02"/>
    <m i="Dwelling_18" v="0"/>
    <m i="Dwelling_19" v="0"/>
    <m i="CREDIT Multi Policy Discount" v="0"/>
    <m i="CREDIT Neighborhood Watch" v="0"/>
    <m i="CREDIT New Home" v="0.20"/>
    <m i="CREDIT New Loan" v="0.10"/>
    <m i="CREDIT Protective Devices" v="0.020000000"/>
    <m i="Dwelling_25" v="0"/>
    <m i="SEC_I C. Personal Property" v="213500"/>
    <m i="SEC_I D. Loss of Use" v="122000"/>
    <m i="SEC_I A. Dwelling" v="305000"/>
    <m i="SEC_II Personal Liability Each Occurrence" v="500000"/>
    <m i="SEC_II Medical Payments Each Person" v="2000"/>
    <m i="SEC_I B. Other Structures" v="30500"/>
    <m i="Total Annual Premium" v="674"/>
    <m i="DED Standard Deductible" v="1000"/>
    <m i="COV Replacement Cost Comp" v="1"/>
    <m i="Total Earthquake Annual Prem" v="0"/>
    <m i="EQ Loss of Use" v="25000"/>
    <m i="EQ Personal Property" v="152500"/>
    <m i="EQ Dwelling Limit" v="305000"/>
    <m i="DED Earthquake Deductible" v="30500"/>
  </c>
  <c i="0">
    <m i="SELECTED_TEIR" v="SPECIAL"/>
    <m i="Total Policy Premium" v="674"/> </c>
</program>
</result>

```

The same nodes are present in the output XML as were found in the input XML; however their meanings are slightly different. The following sections describe how to interpret the output XML.

### <result> Node

The <result> node marks the beginning of a rate request result. There is a one-to-one correspondence between <rate> nodes in the request XML to <result> nodes in the response XML. The result node attributes are defined as follows:

- lob - line of business indicator corresponding to the <rate> request nodes lob attribute.
- gen\_date - server creation timestamp indicating when this response was created.



As described previously, tracking attributes on the <rate> node are returned in the result XML document as attributes in the <result> node.

Example:

```
<rate lob="2" policyId="A1206" gen_date=""2/9/2010 1:50:31 PM">
```

### <program> Node

The result <program> node provides an XML envelope containing all of the formatted data, setup in the RateManager application as output results for the program. There can be (1-N) <program> node groups based on (1-N) program node groups requested in the input <rate> XML document. If the original <rate> request contained multiple <program> nodes in the heading node, each program version located during execution will generate a <program> node in the result data. The result XML program node attributes are defined as follows:

parent_id	- Parent or Insbridge Company Identifier	(available by default)
program_id	- Selected Program Identifier	(available by default)
ver	- Selected Program Version Identifier	(available by default)
Status	- Status of program rate request	(available by default)
company_nm	- Parent Corporate name	(with AddHeading request option)
program_nm	- Program name	(with AddHeading request option)
version_nm	- Version name	(with AddHeading request option)

Example base:

```
<program parent_id='500' program_id='3' ver='2' status='PASS'>
```

Example with “AddHeading” option requested:

```
<program parent_id='200' program_id='32' ver='4' company_nm='NewCo Mutual'
program_nm='Texas 6 Month' ver_nm='Performance' status="PASS">
```

### <c> Node (Category)

The <c> (category) node marks the beginning of output data for a specific category of information (i.e.: home, car, driver, policy, etc). It has the following attributes:

- i - identifier. Zero always indicates “Policy” level inputs, other categories are user defined.  
(available by default)
- d - description of category (with AddResultDesc request option)

Example:

```
<c i="0" d="Policy">
```

The “Policy” category is always a direct child of the <program> node. It is also typically the top-level category node. The Policy category node typically has an ID of zero (i.e.: i=“0”). Item level (user defined) categories are nested under the Policy category node.

Example:

```
<c i="0">
  <m i="SELECTED_TEIR" v="SPECIAL"/>
  <m i="Total Policy Premium" v="674"/>
  <c i="5">
    <m i="Dwelling_1" v="640"/>
    <m i="Dwelling_3" v="0"/>
    <m i="Dwelling_5" v="0"/>
    <m i="SEC_II Personal Liability Each Occurrence" v="500000"/>
    <m i="SEC_II Medical Payments Each Person" v="2000"/>
  </c>
</c>
```

### <m> Node (Map)

The <m> (map) node represents an individual attribute-value pair mapped for a specific SoftRater Package hosted by SoftRater. In the output case, it identifies an output attribute as defined in the SoftRater Package (represented by the <program> node) and its associated value. The <m> node's attributes are defined as follows:

- i - output identifier (available by default)
- n - (name) description of output (with AddResultDesc request option)
- v - value of output (available by default)

The <m> node is always a child of a <c> (category) node and is an attribute-value pair for that specific category instance (see the category example).

### <rate> Node (Input Document)

The <rate> node is optional. When the rate request is issued with the "AddInputs" option, this node is returned in the result XML doc. It includes the full input rate request document that was used to generate the rate result document.

Example:

```
<result lob="1" policyId="ABC1234-AUTO" gen_date="2/9/2010 1:50:31 PM">
  <rate policyId="ABC1234-AUTO">
    ....
  </rate>
  ...
</result>
```

### Input Overrides

By specifying input values in the <program> node within the <header> section of the input XML, those values will be used for that program when it is processed by SoftRater, regardless of whether those values are present in the body of the XML request. This allows each program found in the <header> to use the common set of inputs provided in the rate request body, and either provide additional inputs that

are relevant only to that program, or provide overriding inputs values to ones found in the body, for use during rating.

Example:

```
<rate lob="1">
  <heading>
    <program parent_id="2" program_id="1" program_ver="3">
      <c i="0" d="Policy">
        <m i="11" n="Policy Program Specific Something" v="1029"/>
        <m i="12" n="Custom Question 1" v="XYZ"/>

          <c i="3" d="driver">
            <m i="2" n="gender" v="Female"/>
            <m i="3" n="Custom Driver Question 1" v="ABC"/>
          </c>
        </c>
      </program>
      <program parent_id="2" program_id="7" program_ver="3">
    </heading>
    <c i="0" d="Policy">
      <m i="11" n="Policy Program Specific Something " v="5000"/>
      <c i="3" d="driver">
        <m i="3" n="Custom Driver Question 1" v="DEF"/>
      </c>
    </c>
    .....
  </rate>
```

## Time Statistics

Time tracking statistics can be included if enabled on the Insbridge Framework Administrator, SoftRater Engine page. A time node segment will be included in the Insbridge Response XML document returned from the engine.

Example:

```
<stats>
  <start_time>02/06/2010 04:25:35:0280 PM</start_time>
  <stop_time>02/06/2010 04:25:35:0316 PM</stop_time>
  <running_time>36</running_time>
</stats>
```

### <start\_time>

The <start\_time> is the internal system tracking time from the just before the SoftRaterEJB engine starts any processing, parsing or any manipulation of the Insbridge XML Request but after the XML payload has been marshaled from the integrate client to the SoftRaterEJB system.

### <stop\_time>

The <stop\_time> is the internal system tracking time after all program execution and just before the SoftRaterEJB engine closes the Insbridge Response XML document which will be marshaled back to the integrating client.

### <running\_time>

The <running\_time> is the different (in Milliseconds) from the <start\_time> and <stop\_time>. It represents the transactional duration of the program processing the request.

---

**NOTE:** *Running Time is shown in milliseconds (10<sup>-2</sup> seconds).*

---

## Examples

### Single Rate Request

See Insbridge.XML Request Format and Insbridge.XML Result Format.

### Multiple Rate Requests in a Single XML Document)

It is possible to request several rates from a single XML document. These can be rates on different lines of business, across different states, and/or different SoftRater Package versions.

This simple way to do this is to combine multiple <rate> request nodes in one single root node, and submit it for rating. The root node can be anything, however in the WSI call it is always <ibdoc>.

### Multiple <rate> nodes

This request XML:

```
<ibdoc>
  <rate> ... </rate>
  <rate> ... </rate>
</ibdoc>
```

...will produce this result XML:

```
<ibdoc>
  <result> ... </result>
  <result> ... </result>
</ibdoc>
```

The root node is not returned in the result XML automatically. It must be specified in the call using the “AddRoot” attribute.

---

**NOTE:** *Results may not be returned in the order in which they were submitted. To assure that results are returned in the same order as entered, verify the Web Service Rating Thread entry on the SoftRater Engine page of the Insbridge Framework Administrator (IBFA). An entry of 1 will return results in the same order. An entry greater than the number of rates submitted will return results in order. If you know that you will always have three rates per request, you can set the threads to 3 or greater. Be aware that a higher thread count may affect performance. The default setting is 2.*

---

### Multiple <program> nodes

This request XML:

```

<ibdoc>
  <rate>
    <heading>
      <program> ... </program>
      <program> ... </program>
    </heading >
    <c>...</c>
  </rate>
</ibdoc>

```

...will produce this result XML:

```

<ibdoc>
  <result>
    <program>
      <c>...</c>
    </program>
    <program>
      <c>...</c>
    </program>
  </result>
</ibdoc>

```

### Multi - State Request

To rate against multiple states using one request XML document, it is recommended to follow the “Multiple <program> nodes” request model. In the following example, we are targeting two Auto programs for rating. A program typically represents a State for a specific line of business. In this example we will assume CA=“21” and TX=“41”. As discussed earlier in Input Overrides, each program entry can specify input values to be used for that particular program.

This rate request XML:

```

<ibdoc>
  <rate lob="1">
    <heading>
      <program parent_id="2" program_id="21"> ... </program>
      <program parent_id="2" program_id="41"> ... </program>
    </heading >
    <c>...</c>
  </rate>
</ibdoc>

```

... will produce this result XML:

```

<ibdoc>
  <result lob="1" gen_date="2/9/2010 1:50:31 PM">
    <program parent_id="2" program_id="21">
      <c>...</c>
    </program>
    <program parent_id="2" program_id="41">
      <c>...</c>
    </program>
  </result>
</ibdoc>

```

The diagram shows two green boxes on the right side of the XML output. The top box is labeled "Results for CA" and has an arrow pointing to the first program element: `<program parent_id="2" program_id="21">`. The bottom box is labeled "Results for TX" and has an arrow pointing to the second program element: `<program parent_id="2" program_id="41">`.

```
</result>  
</ibdoc>
```

## Multi - Line of Business Request

To rate against multiple lines of business using one request XML document, it is recommended to follow the "Multiple <rate> nodes" request model. In the following example, we are targeting two lines of business for rating, Auto and Home. The "lob" attribute in the <rate> node signifies which line of business will be rated against.

This rate request XML:

```
<ibdoc>  
  <rate lob="1" > ... </rate>  
  <rate lob="2" > ... </rate>  
</ibdoc>
```

...will produce this result XML:

```
<ibdoc>  
  <result lob="1" gen_date="2/9/2010 1:50:31 PM"> ... </result>  
  <result lob="2" gen_date="2/9/2010 1:50:45 PM"> ... </result>  
</ibdoc>
```

Results for Auto

Results for Home

# SOFTWARE INTEGRATION FOR JAVA

SoftRater is an EJB component hosted in the Application Server and accessible through the following software integration methods. Each Application Server has a default port that is used.

- **WebLogic:** Port 7001
- **WebSphere:** Port 9080
- **JBoss:** Port 8080

---

**NOTE:** *PORT# will change depending on your Application Server. These defaults are current as of the IBRU 4.5.1 Release.*

---

When rating custom XML, the engine has the option of stateful rating. The SoftRater WSI will add the contents of the rating results to the document that was submitted for rating. This is an important consideration when constructing XSLT (mapping) files.

If you are submitting Insbridge XML, it is recommended that you use HTTP POST instead of SOAP. SOAP should only be used when rating custom XML.

## Software Integration Methods

1. **HTTP SOAP Proxy** – SoftRater Web Service – WSDL documents can be found in IBSS. From the WSDL documents, proxy classes can be generated to communicate with the installed SoftRater instance. The SoftRater Web Service WSDL document and a sample SoftRater SOAP proxy class instance are located in [IBSS → SoftRater Version 3 → Get Sample Files](#).

Whether the sample SoftRaterProxy class is used or a new one is generated from the WSDL interface, the integrating application must call the ([setEndPoint](#)) method to set the target hosting URL for the SoftRater instance. SOAP end point information is located in IBSS, [IBSS → SoftRater Version 3 → Get WSDL \(url\)](#) and [IBSS → SoftData Version 1 → Get WSDL \(url\)](#).

2. **HTTP POST** – A lite weight Web Service Interface JSP page is provided as an interface for clients with just web form POST abilities. Custom XML cannot be submitted from this interface.

The URL to the POST interface should be the following:  
<http://<yourserver>:PORT#/IBSS/softrater/lwsi/RateBase.jsp>

3. **EJB** – Direct JNDI interfacing.

The EJB interfaces for creating service clients, which creates SoftRater instances.

- WebSphere JNDI Path – [ejb/com/insbridge/softrater/SoftRaterHome](#)
- WebLogic JNDI Path – [ejb.SoftRaterEJBRemoteHome](#)
- JBoss JNDI Path – [IBSS/SoftRaterEJB/local](#)
  
- Target JAR – [InsBridgeEJB.jar](#) is part of the [IBSS\\_xx.EAR](#).

- WebSphere JNDI Path – [ejb/com/insbridge/sofrater/SoftRaterHome](#)
- WebLogic JNDI Path – [ejb.SoftRaterEJBRemoteHome](#)
- JBoss JNDI Path – [IBSS/SoftRaterEJB/local](#)

**For WebSphere ONLY:** Target JAR contains the following standard EJB 2.0 interface files.

- SoftRater – Remote interface for Enterprise Bean
- SoftRaterBean – Bean implementation class for Enterprise Bean
- SoftRaterHome – Home interface for Enterprise Bean

### Interface Example:

```

package com.insbridge.sofrater;
import java.rmi.RemoteException;

/**
 * Remote interface for Enterprise Bean: SoftRater
 */
public interface SoftRater extends javax.ejb.EJBObject
{
    public String ProcessIB(                final String XMLInputs,
                                           final boolean AddRootNode,
                                           final boolean AddInputsNode,
                                           final boolean AddHeadingNode,
                                           final boolean AddResultDesc,
                                           final boolean AddResultThatAreEmpty,
                                           final boolean DoInStyle,
                                           final boolean DoDebugOutput
                                           )

    throws RemoteException;

    public boolean ResetPackageCache( ) throws RemoteException;

    public void ResetConfigInfo( ) throws RemoteException;

    public String QueryAvailableEnvironments( ) throws RemoteException;

    public String getVersion( ) throws RemoteException;

    public String getErrorMessage( ) throws RemoteException;

    public String getDefaultPath() throws RemoteException;
}

```



## RATING ARGUMENTS FOR JAVA

The SoftRater engine rating arguments control the handling of XML data out of the system. Rating arguments are optional. For optimal performance, use the following arguments for your rating integration.

- **Add Root Node (Use default – False)** – If submitting multiple rate request documents, this option is typically set to true to make the result document a valid XML document.
- **Add Inputs (Use default – False)** – When set to true, the full request Insbridge.XML document is returned in the result Insbridge.XML document making the XML document much larger than normal.
- **Add Heading (Use default – False)** – When set to true, the program name description information is returned in the result XML also.
- **Add Result Descriptions (Use default – False)** – When set to true each result item includes the RateManager variable result name along with the result id and value. Making the result XML much larger. Typically, most integration operates on the result IDs and descriptions are not needed when building an automated system.
- **Add Empty Results (Use default – False)** – When set to true, a defined result item, whose value is empty (i.e. blank), is still created and returned blank in the resulting Insbridge.XML. If your program design requires a number of optional results, you could have blank results items in your XML.
- **Debug Output (Use default – False)** – When set to false, no debug report will be issued. Set to true if you would like a debug report.
- **Encode (Use default – True)** – When set to true, encoding will be done to special characters. Set to false if you would like to not use encoding. Entering a value here will override any value entered in the file. No encoding may result in errors if special characters are submitted in the XML.
- **EnvRef (Use default – blank)** – When left blank, the default environment defined in setup will be used. To specifically define an environment, enter in the environment name. Entering a value here will override any value entered in the file.

If no arguments are sent, the default values will be used. Default values are defined in setup. If using SoftRater Server, see Submit Insbridge XML in the IBSS User Guide. To navigate to the **Submit Insbridge XML** page, select the Submit Insbridge (XML) link available from the menu tree on the SoftRater page of IBSS. This page allows you to rate an input file that is in the Insbridge.XML format.

## CUSTOM XML ARGUMENTS FOR JAVA

The SoftRater WSI controls the processing (transformations) of XML data in and/or out of the system. Custom XML arguments are required only when you are submitting custom XML. If you are using Insbridge XML, custom XML arguments are not required. The information will be contained in the Insbridge XML. If you are using custom XML and do not define the custom XML arguments, any error message will be thrown.

Use the following MapRequest SOAP options below for your rating integration.

- Lob – Identifier of the Line Of Business
- Subscriber – Identifier of the Subscriber
- Program – Identifier of the Program
- Version – Identifier of the Program Version
- InputMappingIdentifier – Name of the Custom Mapping Document
- InputMappingType – Integer for the Custom Mapping Document
  - 0 = None, no input mapping should be performed
  - 1 = Global, input mapping is global to the Line of Business. Mapping name required
  - 2 = Local, input mapping is unique to the program version
  - 3 = Custom, input mapping of the customer that has been added into the workflow. Mapping name required
- OutputMappingIdentifier – Name of the Custom Mapping Document
- OutputMappingType – Integer for the Custom Mapping Document
  - 0 = None, no output mapping should be performed
  - 1 = Global, output mapping is global to the Line of Business
  - 2 = Local, output mapping is unique to the program version
  - 3 = Custom, output mapping of the customer that has been added into the workflow
- OutputMappingStateful – The SoftRater WSI will add the contents of the rating results to the document that was submitted for rating.
- OutputErrorXPathLoc – Location of any system errors that occurred during the Webservice request that are not related to SoftRater. (By default, an error node is created at the root level.)
- UseResultEncoding – setting should be set to 1. This will allow encoding. 0 = no encoding. No encoding may result in errors if special characters are submitted in the XML.

---

**NOTE:** *If you are using custom XML to rate or test, the mapping name may need to be passed through. The Input Mapping Type arguments **Global** and **Custom** require the name of the mapping file.*

---

**NOTE:** *The **OutputSchema** web services argument is no longer being used. This argument displayed the path of any schema that the WSI should validate against. If you are currently using this, you can leave it in the custom XML.*

---

### INSBRIDGE.XML JAVA EXAMPLE

XML is the primary data exchange mechanism used by Oracle Insurance Insbridge Rating and Underwriting system to communicate information electronically with external and internal software systems.

Insbridge's rating request input XML is designed to be flexible and efficient. It allows for single or multiple rate requests to be submitted via one input XML document. The rate requests embedded in this single document can be targeted to multiple states and/or multiple lines of business. Multiple versions of a rating package also can be targeted in a single rate request document.

The rating request response XML is also streamlined to present all the results to the various request methods, described above, in a single output XML document.

### INSBRIDGE.XML REQUEST FORMAT

The following is an example of an Insbridge rate request XML document:

```
<rate lob="2" tracking_attribute="" env_def="">
  <heading>
    <program parent_id="700" program_id="24" program_ver="1"/>
  </heading>
  <c i="0" desc="Policy">
    <m i="1086" n="PackageDiscInd" v=""/>
    <m i="1094" n="RenewalRetentionCreditInd" v=""/>
    <m i="1157" n="CompanyCode" v=""/>
    <m i="1212" n="Eff_Date" v=""/>
    <m i="1214" n="PrimInsuredAge" v=""/>
    <m i="1215" n="SecInsuredAge" v=""/>
    <m i="1222" n="RenewalInd" v=""/>
  <c i="5" desc="Home">
    <m i="1083" n="TerritoryCode" v=""/>
    <m i="1084" n="ResidenceType" v=""/>
    <m i="1087" n="ProtectionClass" v=""/>
    <m i="1095" n="Wood/Tile/SlateRoofType" v=""/>
    <m i="1096" n="HomeDeductible" v=""/>
    <m i="1098" n="WindstormOrHailDeductible" v=""/>
    <m i="1100" n="CentralStationFireAlarmInd" v=""/>
    <m i="1101" n="CentralStationBurglarAlarmInd" v=""/>
  <c i="8" desc="Coverage">
    <m i="1204" n="CovCd" v=""/>
    <m i="1205" n="CovLimit" v=""/>
    <m i="1207" n="CovEff_Date" v=""/>
  </c>
  <c i="9" desc="Endorsement">
    <m i="1181" n="EndorCd" v=""/>
    <m i="1182" n="EndorRateInd" v=""/>
    <m i="1190" n="EndorEff_Date" v=""/>
  </c>
</rate>
```

```

        <m i="1191" n="Parm5" v=""/>
      </c>
    </c>
  </c>
</rate>

```

## <rate> Node

The <rate> node marks the beginning of a rate request for a specific line of business. This node has the required attribute, “lob”, which identifies the “line of business” for the request. In the following example, the lob attribute is set to “2” which represents “Home” insurance according to the Insbridge standard line of business (see RateManager Admin User Guide). The rate node attributes are defined as follows:

<b>lob</b>	– line of business indicator per Insbridge standard lob codes (Required)
<b>env_def</b>	– Allows the user to override the default rating environment. By default, SoftRater rates against the default environment, as set up in the Insbridge Framework Administrator (see Introduction to Environments in the Insbridge Framework Administrator User Guide). To rate against a different environment, add the attribute env_def="Env_Name" to the rate node, where Env_Name is the name of the environment you wish to rate against.

Example:

```
<rate lob="1" env_def="Env_Name">
```

<b>renc</b>	– Allows the user to instruct SoftRater to not encode XML characters that are not considered valid XML characters. These characters are: <ul style="list-style-type: none"> <li>- Ampersand (&amp;)</li> <li>- Less than sign (&lt;)</li> <li>- Greater than sign (&gt;)</li> <li>- Double quotation marks (")</li> <li>- Single quotation mark (')</li> </ul>
-------------	--

By default, these characters are encoded in the result XML. For example, the ampersand is encoded as &amp;. To override this default behavior, add the attribute renc="1" to the rate node.

Example:

```
<rate lob="1" renc="1">
```

As an optional feature, all other attributes provided on the <rate> node are collected as tracking attributes to be returned in the result XML document as attributes in the <result> node. This allows the original rate request to be uniquely tracked with its result XML document by any identification elements available to the calling subsystem. In the example below, the “ policyId='A1206' ” attribute value pair would be mirrored on the <result> node of the resulting output XML.

Example:

```
<rate lob="2" policyId="A1206">
```

This rate request may be targeted to one or more rating logic instances based on what is found in the <heading> node.

## <heading> Node

The <heading> node serves only as a container for <program> nodes and has no attributes. If multiple <program> nodes are found in the heading node, then rating is performed for each node, if possible, and appropriate results are generated in the output XML.

Example:

```
<heading>
  <program parent_id="700" program_id="24"
    program_ver="1"/>
  <program parent_id="700" program_id="22"
    program_ver="1"/>
</heading>
```

## <program> Node

The <program> node specifies a specific SoftRater Package (rating engine logic instance) to run this rate request against. A program typically represents rating logic for a particular State and line of business (e.g.: Texas Auto insurance, California Home insurance). The program node attributes are defined as follows:

- parent\_id - Insbridge identifier typically assigned to a carrier for database lookup purposes (Required)
- program\_id - Insbridge identifier assigned to a program (rating engine logic instance) which represent the rating rules necessary to generate a quote. (Optional)
- program\_ver - A particular version of a program. Each version may have different rating rules, inputs, outputs, etc. (Optional)

### Attribute Requirement Rules:

1. If the [program\\_id](#) is provided then the version [program\\_ver](#) value is used to select a program version to be processed against the source XML.  
Example: <program parent\_id="100" program\_id="100" ver="3"/>
2. If the [program\\_id](#) is provided and the [program\\_ver](#) is not the version selection rules for the program, identified by program\_id, are used to select the program version to be processed against the source XML.  
Example: <program parent\_id="100" program\_id="100" />

The <program> node allows mapped input overrides to be specified for a specific program in a multi-program (or multi-state) rate request. See Multiple Rate Requests in a Single XML Document for more information.

### <program> node

**result\_def** – Allows the user to override the default result mapping group. By default, SoftRater uses the group that is set up as the default in RateManager. To use a different result group, add the attribute result\_def="Result\_Code" to the rate node, where Result\_Code is the result code shown in RateManager.

```
<program parent_id="123" program_id="1" program_ver="1" result_def="CA7E730716"/>
```

Group Name	Last Updated	Enable	Default	D/V Output Separate	Result Code
Austin1	Mar 15 2010 11:46AM	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	CA7E730716

Figure 2 SoftRater Switches

In order for you to be able to use a result group, it must have been **Enabled** when the package was created (see the RateManager topic Editing Result Group Options).

### <c> Node (Category)

The <c> (category) node marks the beginning of input data for a specific category of information (i.e.: home, car, driver, policy, etc). It has the following attributes:

- i - identifier. Zero always indicates “Policy” level inputs, other categories are user definable. (Required)
- desc - description of category (Optional)

Example:

```
<c i="0" desc="Policy">
```

The “Policy” category is always a direct child of the <rate> node, except for overrides in the <program> node (described later). It is also the top-level category node. The Policy category node typically has an ID of zero (i.e.: i="0"). Item level categories are nested under the Policy category node.

Example:

```
<c i="0" desc="Policy">
  <m i="1212" n="Eff_Date" v="" />
  <m i="1214" n="PrimInsuredAge" v="" />
  <m i="1215" n="SecInsuredAge" v="" />
  <c i="5" desc="Home">
    ....
    <c i="7" desc="Scheduled Property">
      ....
    </c>
  </c>
  <c i="5" desc="Home">
    ....
  </c>
</c>
```

### <m> Node (Map)

The <m> (map) node represents an individual attribute-value pair mapped for a specific SoftRater Package hosted by SoftRater. In the input case, it identifies an input attribute recognized by the SoftRater

Package (or Packages) listed in the <heading> node, and its associated value. The <m> node's attributes are defined as follows:

- i - input identifier (Required)
- n - name, description of input (Optional)
- v - value of input (Required)

The <m> node is always a child of a <c> (category) node and is an attribute-value pair for that specific category instance.

## INSBRIDGE.XML RESULT FORMAT

The following is an example of an Insbridge rate request result XML document. The result XML is very similar to the input XML format.

```
<result lob="2" st="6" gen_date="2/19/2010 1:55:32 PM">
  <program parent_id="900" program_id="1" program_ver="1" status="PASS">
    <c i="5">
      <m i="Dwelling_1" v="640"/>
      <m i="Dwelling_3" v="0"/>
      <m i="Dwelling_5" v="0"/>
      <m i="replcc" v="0"/>
      <m i="COV Replacement Cost Contents" v="0"/>
      <m i="COV Mortgage Payment" v="0"/>
      <m i="Dwelling_10" v="0"/>
      <m i="Dwelling_11" v="34"/>
      <m i="COV Replacement Cost - Dwelling" v="0"/>
      <m i="Dwelling_13" v="0"/>
      <m i="Dwelling_13" v="674"/>
      <m i="COV Business Pursuits" v="0"/>
      <m i="COV Permitted Incidental Occupancies" v="0"/>
      <m i="COV Personal Injury" v="0"/>
      <m i="Dwelling_17" v="0.02"/>
      <m i="Dwelling_18" v="0"/>
      <m i="Dwelling_19" v="0"/>
      <m i="CREDIT Multi Policy Discount" v="0"/>
      <m i="CREDIT Neighborhood Watch" v="0"/>
      <m i="CREDIT New Home" v="0.20"/>
      <m i="CREDIT New Loan" v="0.10"/>
      <m i="CREDIT Protective Devices" v="0.020000000"/>
      <m i="Dwelling_25" v="0"/>
      <m i="SEC_I C. Personal Property" v="213500"/>
      <m i="SEC_I D. Loss of Use" v="122000"/>
      <m i="SEC_I A. Dwelling" v="305000"/>
      <m i="SEC_II Personal Liability Each Occurrence" v="500000"/>
      <m i="SEC_II Medical Payments Each Person" v="2000"/>
      <m i="SEC_I B. Other Structures" v="30500"/>
      <m i="Total Annual Premium" v="674"/>
      <m i="DED Standard Deductible" v="1000"/>
      <m i="COV Replacement Cost Comp" v="1"/>
      <m i="Total Earthquake Annual Prem" v="0"/>
      <m i="EQ Loss of Use" v="25000"/>
      <m i="EQ Personal Property" v="152500"/>
    </c>
  </program>
</result>
```

```

        <m i="EQ Dwelling Limit" v="305000"/>
        <m i="DED Earthquake Deductible" v="30500"/>
    </c>
    <c i="0">
        <m i="SELECTED_TEIR" v="SPECIAL"/>
        <m i="Total Policy Premium" v="674"/> </c>
    </program>
</result>

```

The same nodes are present in the output XML as were found in the input XML, however their meanings are slightly different.

### <result> Node

The <result> node marks the beginning of a rate request result. There is a one-to-one correspondence between <rate> nodes in the request XML to <result> nodes in the response XML. The result node attributes are defined as follows:

lob - line of business indicator corresponding to the <rate> request nodes lob attribute  
gen\_date - server creation timestamp indicating when this response was created.

The tracking attributes on the <rate> node are returned in the result XML document as attributes in the <result> node.

Example:

```
<rate lob="2" policyId="A1206" gen_date="2/9/2010 1:50:31 PM">
```

### <program> Node

The result <program> node provides an XML envelope containing all of the formatted data, setup in the RateManager application as output results for the program. There can be (1-N) <program> node groups based on (1-N) program node groups requested in the input <rate> XML document. If the original <rate> request contained multiple <program> nodes in the heading node, each program version located during execution will generate a <program> node in the result data. The result XML program node attributes are defined as follows:

parent_id	- Parent or Insbridge Company Identifier	(available by default)
program_id	- Selected Program Identifier	(available by default)
ver	- Selected Program Version Identifier	(available by default)
Status	- Status of program rate request	(available by default)
company_nm	- Parent Corporate name	(with AddHeading request option)
program_nm	- Program name	(with AddHeading request option)
version_nm	- Version name	(with AddHeading request option)

Example base:

```
<program parent_id='500' program_id='3' ver='2' status='PASS'>
```

Example with add heading option requested:

```
<program parent_id='200' program_id='32' ver='4' company_nm='NewCo Mutual'
program_nm='Texas 6 Month' ver_nm='Performance' status="PASS">
```



### <c> Node (Category)

The <c> (category) node marks the beginning of output data for a specific category of information (i.e.: home, car, driver, policy, etc). It has the following attributes:

- i - identifier. Zero always indicates “Policy” level inputs, other categories are user defined. (available by default)
- d - description of category (with AddResultDesc request option)

Example:

```
<c i="0" d="Policy">
```

The “Policy” category is always a direct child of the <program> node. It is also typically the top level category node. The Policy category node typically has an ID of zero (i.e.: i=“0”). Item level (user defined) categories are nested under the Policy category node.

Example:

```
<c i="0">
  <m i="SELECTED_TEIR" v="SPECIAL"/>
  <m i="Total Policy Premium" v="674"/>
  <c i="5">
    <m i="Dwelling_1" v="640"/>
    <m i="Dwelling_3" v="0"/>
    <m i="Dwelling_5" v="0"/>
    <m i="SEC_II Personal Liability Each Occurrence" v="500000"/>
    <m i="SEC_II Medical Payments Each Person" v="2000"/>
  </c>
</c>
```

### <m> Node (Map)

The <m> (map) node represents an individual attribute-value pair mapped for a specific SoftRater Package hosted by SoftRater. In the output case, it identifies an output attribute as defined in the SoftRater Package (represented by the <program> node) and its associated value. The <m> node's attributes are defined as follows:

- i - output identifier (available by default)
- n - (name) description of output (with AddResultDesc request option)
- v - value of output (available by default)

The <m> node is always a child of a <c> (category) node and is an attribute-value pair for that specific category instance (see the category example above).

### <rate> Node (Input Document)

The <rate> node is optional. When the rate request is issued with the “AddInputs” option, this node is returned in the result XML document. It includes the full input rate request document that was used to generate the rate result document.

Example:

```
<result lob="1" policyId="ABC1234-AUTO" gen_date="2/9/2010 1:50:31 PM">
    <rate policyId="ABC1234-AUTO">
        ....
    </rate>
    ...
</result>
```

## Input Overrides

By specifying input values in the <program> node within the <header> section of the input XML, those values will be used for that program when it is processed by SoftRater, regardless of whether those values are present in the body of the XML request. This allows each program found in the <header> to use the common set of inputs provided in the rate request body, and either provide additional inputs that are relevant only to that program, or provide overriding inputs values to ones found in the body, for use during rating.

Example:

```
<rate lob="1">
  <heading>
    <program parent_id="2" program_id="1" program_ver="3">
      <c i="0" d="Policy">
        <m i="11" n="Policy Program Specific Something " v="1029"/>
        <m i="12" n="Custom Question 1" v="XYZ"/>
      </c>
      <c i="3" d="driver">
        <m i="2" n="gender" v="Female"/>
        <m i="3" n="Custom Driver Question 1" v="ABC"/>
      </c>
    </program>
    <program parent_id="2" program_id="7" program_ver="3"/>
  </heading>
  <c i="0" d="Policy">
    <m i="11" n="Policy Program Specific Something " v="5000"/>
    <c i="3" d="driver">
      <m i="3" n="Custom Driver Question 1" v="DEF"/>
    </c>
  </c>
  ....
</rate>
```

## Time Statistics

Time tracking statistics can be included if enabled on the Insbridge Framework Administrator, SoftRater Engine page. The following node segment will be included in the Insbridge Response XML document returned from the engine.

Example:

```
<stats>
  <start_time>02/06/2010 04:25:35:0280 PM</start_time>
```

```
<stop_time>02/06/2010 04:25:35:0316 PM</stop_time>
<running_time>36</running_time>
</stats>
```

#### <start\_time>

The <start\_time> is the internal system tracking time from the just before the SoftRaterEJB engine starts any processing, parsing or any manipulation of the Insbridge XML Request but after the XML payload has been marshaled from the integrate client to the SoftRaterEJB system.

#### <stop\_time>

The <stop\_time> is the internal system tracking time after all program execution and just before the SoftRaterEJB engine closes the Insbridge Response XML document which will be marshaled back to the integrating client.

#### <running\_time>

The <running\_time> is the different (in Milliseconds) from the <start\_time> and <stop\_time>. It represents the transactional duration of the program processing the request.

---

**NOTE:** *Running Time is shown in milliseconds (10<sup>-2</sup> seconds).*

---

## Examples

### Single Rate Request

#### Insbridge.XML Request Format.

```
<rate lob="2" tracking_attribute="" env_def="">
  <heading>
    <program parent_id="700" program_id="24" program_ver="1"/>
  </heading>
  <c i="0" desc="Policy">
    <m i="1086" n="PackageDiscInd" v=""/>
    <m i="1094" n="RenewalRetentionCreditInd" v=""/>
    <m i="1157" n="CompanyCode" v=""/>
    <m i="1212" n="Eff_Date" v=""/>
    <m i="1214" n="PrimInsuredAge" v=""/>
    <m i="1215" n="SecInsuredAge" v=""/>
    <m i="1222" n="RenewalInd" v=""/>
    <c i="5" desc="Home">
      <m i="1083" n="TerritoryCode" v=""/>
      <m i="1084" n="ResidenceType" v=""/>
      <m i="1087" n="ProtectionClass" v=""/>
      <m i="1095" n="Wood/Tile/SlateRoofType" v=""/>
      <m i="1096" n="HomeDeductible" v=""/>
      <m i="1098" n="WindstormOrHailDeductible" v=""/>
      <m i="1100" n="CentralStationFireAlarmInd" v=""/>
      <m i="1101" n="CentralStationBurglarAlarmInd" v=""/>
    </c>
    <c i="8" desc="Coverage">
      <m i="1204" n="CovCd" v=""/>
      <m i="1205" n="CovLimit" v=""/>
      <m i="1207" n="CovEff_Date" v=""/>
    </c>
  </c>
```

```

    <c i="9" desc="Endorsement">
      <m i="1181" n="EndorCd" v=""/>
      <m i="1182" n="EndorRateInd" v=""/>
      <m i="1190" n="EndorEff_Date" v=""/>
      <m i="1191" n="Parm5" v=""/>
    </c>
  </c>
</c>
</rate>

```

### Insbridge.XML Result Format.

```

<result lob="2" st="6" gen_date="2/19/2010 1:55:32 PM">
  <program parent_id="900" program_id="1" program_ver="1" status="PASS">
    <c i="5">
      <m i="Dwelling_1" v="640"/>
      <m i="Dwelling_3" v="0"/>
      <m i="Dwelling_5" v="0"/>
      <m i="replcc" v="0"/>
      <m i="COV Replacement Cost Contents" v="0"/>
      <m i="COV Mortgage Payment" v="0"/>
      <m i="Dwelling_10" v="0"/>
      <m i="Dwelling_11" v="34"/>
      <m i="COV Replacement Cost - Dwelling" v="0"/>
      <m i="Dwelling_13" v="0"/>
      <m i="Dwelling_13" v="674"/>
      <m i="COV Business Pursuits" v="0"/>
      <m i="COV Permitted Incidental Occupancies" v="0"/>
      <m i="COV Personal Injury" v="0"/>
      <m i="Dwelling_17" v="0.02"/>
      <m i="Dwelling_18" v="0"/>
      <m i="Dwelling_19" v="0"/>
      <m i="CREDIT Multi Policy Discount" v="0"/>
      <m i="CREDIT Neighborhood Watch" v="0"/>
      <m i="CREDIT New Home" v="0.20"/>
      <m i="CREDIT New Loan" v="0.10"/>
      <m i="CREDIT Protective Devices" v="0.020000000"/>
      <m i="Dwelling_25" v="0"/>
      <m i="SEC_I C. Personal Property" v="213500"/>
      <m i="SEC_I D. Loss of Use" v="122000"/>
      <m i="SEC_I A. Dwelling" v="305000"/>
      <m i="SEC_II Personal Liability Each Occurrence" v="500000"/>
      <m i="SEC_II Medical Payments Each Person" v="2000"/>
      <m i="SEC_I B. Other Structures" v="30500"/>
      <m i="Total Annual Premium" v="674"/>
      <m i="DED Standard Deductible" v="1000"/>
      <m i="COV Replacement Cost Comp" v="1"/>
      <m i="Total Earthquake Annual Prem" v="0"/>
      <m i="EQ Loss of Use" v="25000"/>
      <m i="EQ Personal Property" v="152500"/>
      <m i="EQ Dwelling Limit" v="305000"/>
      <m i="DED Earthquake Deductible" v="30500"/>
    </c>
    <c i="0">
      <m i="SELECTED_TEIR" v="SPECIAL"/>
    </c>
  </program>
</result>

```

```
        <m i="Total Policy Premium" v="674"/> </c>
    </program>
</result>
```

## Multiple Rate Requests in a Single XML Document

It is possible to request several rates from a single XML document. These can be rates on different lines of business, across different states, and/or different SoftRater Package versions.

You can do this by combining multiple <rate> request nodes in one single root node, and submitting it for rating. The root node can be anything, however in the WSI call it is always <ibdoc>.

### Multiple <rate> nodes

This request XML:

```
<ibdoc>
  <rate> ... </rate>
  <rate> ... </rate>
</ibdoc>
```

...will produce this result XML:

```
<ibdoc>
  <result> ... </result>
  <result> ... </result>
</ibdoc>
```

The root node is not returned in the result XML automatically. It must be specified in the call using the "AddRoot" attribute.

### Multiple <program> nodes

This request XML:

```
<ibdoc>
  <rate>
    <heading>
      <program> ... </program>
      <program> ... </program>
    </heading >
    <c>...</c>
  </rate>
</ibdoc>
```

...will produce this result XML:

```
<ibdoc>
  <result>
    <program>
```

```

        <c>...</c>
    </program>
</program>
    <c>...</c>
</program>
</result>
</ibdoc>

```

## Multi - State Request

To rate against multiple states using one request XML document, it is recommended to follow the “Multiple <program> nodes” request model. In the following example, we are targeting two Auto programs for rating. A program typically represents a State for a specific line of business. In this example we will assume CA=“21” and TX=“41”. As discussed earlier in Input Overrides, each program entry can specify input values to be used for that particular program.

This rate request XML:

```

<ibdoc>
  <rate lob="1">
    <heading>
      <program parent_id="2" program_id="21"> ... </program>
      <program parent_id="2" program_id="41"> ... </program>
    </heading>
    <c>...</c>
  </rate>
</ibdoc>

```

... will produce this result XML:

```

<ibdoc>
  <result lob="1" gen_date="2/9/2010 1:50:31 PM">
    <program parent_id="2" program_id="21"> ← Results for CA
      <c>...</c>
    </program>
    <program parent_id="2" program_id="41"> ← Results for TX
      <c>...</c>
    </program>
  </result>
</ibdoc>

```

## Multi - Line of Business Request

To rate against multiple lines of business using one request XML document, it is recommended to follow the “Multiple <rate> nodes” request model. In the following example, we are targeting two lines of business for rating, Auto and Home. The “lob” attribute in the <rate> node signifies which line of business will be rated against.

This rate request XML:

```

<ibdoc>
  <rate lob="1" > ... </rate>
  <rate lob="2" > ... </rate>

```

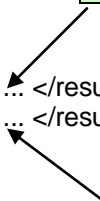
</ibdoc>

...will produce this result XML:

```
<ibdoc>  
  <result lob="1" gen_date="2/9/2010 1:50:31 PM"> ... </result>  
  <result lob="2" gen_date="2/9/2010 1:50:45 PM"> ... </result>  
</ibdoc>
```

Results for Auto

Results for Home



## UPGRADE NOTICE

This notice is for customers currently running a SoftRater for Java engine Release 3.12 or lower. This includes:

- **SoftRater for WebLogic**
- **SoftRater for WebSphere**
- **SoftRater for JBoss**

Necessary updates have been made to the SoftRater for Java engines. This includes:

- **SoftRater for WebSphere has been updated to SOAP version 1.1.**
- **SoftRater for WebLogic has been updated to SOAP version 1.2.**
- **SoftRater for JBoss has been updated to SOAP version 1.2.**

These changes mean that the WSDLs for the SoftRater for Java engines have been modified. Updated WSDL documents will need to be incorporated into any client calling application that communicates with SoftRater. If upgrades are done to the SoftRater for Java engines without updating the calling applications, the calling application will fail.

**Please update a test environment prior to deploying to production.** Install the .EAR file in a test environment to obtain the WSDL. Update your calling application and then test. When testing is complete, you can upgrade your other environments.

---

**NOTE:** *Customers running a Windows only environment, without a Java component, will not be affected.*

---

These updates are for any instance going from 3.12 to a greater of SoftRater for Java engines only.



## **CONTACTING SUPPORT**

If you need assistance with an Oracle Insurance Insbridge Rating and Underwriting System product, please log a Service Request using My Oracle Support at <https://support.oracle.com/>.

### **Address any additional inquiries to:**

**Oracle Corporation**  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[oracle.com](http://oracle.com)

## **TTY ACCESS TO ORACLE SUPPORT SERVICES**

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, seven days a week. For TTY support, call 800.446.2398.

## INDEX

### (

(1-N) <program> node Groups, 24, 39  
(setEndPoint) Method  
    Called by Application, 30

### <

<c> Node, 21, 24, 37, 40  
<heading> Node, 20, 36  
<input> Node, 25, 40  
<m> Node, 22, 25, 37, 40  
<program> node  
    Available Switches, 36  
<program> Node, 21, 24, 36, 39  
    Specifying Input Values, 25, 41  
<rate> Node, 20, 35  
<result> Node, 23, 39  
<running\_time> Node, 26, 42  
<start\_time> Node, 26, 42  
<stop\_time> Node, 26, 42

### A

Add Empty Results  
    Options, 16, 32  
Add Heading  
    Options, 16, 32  
Add Inputs  
    Options, 16, 32  
Add Result Descriptions  
    Options, 16, 32  
Add Root Node  
    Options, 16, 32  
AddInputs Option, 25, 40  
AddRoot  
    Attribute, 27, 44  
Ampersand &  
    Invalid Characters, 35  
Applications  
    Supported by Insbridge, 7  
ASP.NET  
    SoftRater WSI, 12  
Attribute  
    env\_def, 20, 35  
    renc, 35  
    result\_def, 21, 36  
    Rules, 21, 36  
Attributes  
    AddRoot, 27, 44  
    c Node, 21, 24, 37, 40  
    company\_nm, 24, 39  
    d, 24, 40  
    desc, 22, 37

    env\_def, 20, 35  
    gen\_date, 23, 39  
    i, 22, 24, 25, 37, 38, 40  
    Line of Business, 20, 35  
    lob, 20, 23, 35, 39  
    m Node, 22, 37  
    n, 22, 25, 38, 40  
    parent\_id, 21, 36  
    Program Node, 21, 24, 36, 39  
    program\_id, 21, 24, 36, 39  
    program\_ver, 21, 36  
    Rate Node, 20, 35  
    renc, 35  
    Results Node, 23, 39  
    Status, 24, 39  
    Tracking, 20, 35  
    v, 22, 25, 38, 40  
    ver, 24, 39  
    version\_nm, 24, 39  
Attribute-value Pair, 22, 38

### C

C# Example, 14, 15  
Cache  
    Breakdown of, 9  
    Defining a Hit, 9  
    Defining a Miss, 10  
    Disabled, 11  
    Exceptions, 11  
Cache Hit  
    Example, 10  
Cache Miss  
    Example, 10  
Category Node  
    Example, 22, 24, 37, 40  
    Top Level, 22, 24, 37, 40  
Category Node Nested  
    Example, 22, 25, 37, 40  
Children  
    Category Node, 22, 25, 38, 40  
    Rate Node, 22, 24, 37, 40  
company\_nm  
    Attributes, 24, 39  
Custom Message, 15  
CustomXml  
    Submitting, 13, 30

### D

d  
    Attribute, 24, 40  
Data  
    Handling Out of System, 16, 32  
Debug Output, 16

## Defining

- Cache Hit, 9
- Cache Miss, 10

## Definition

- SoftRater, v

## desc

- Attribute, 22, 37

## Direct JNDI interfacing, 30

## Disabled

- Cache, 11

## Double Quote

- Invalid Characters, 35

**E**

## Edition Notice, 2

## EJB 2.0 Interface Files, 31

## EJB Component

- WebSphere, 30

## EJB Interfaces, 30

## Enabled

- Result Group, 21, 37

## env\_def, 35

- Attributes, 20, 35

## Environment

- Web Service Interface, 12

## EnvRef, 16

## Error Node

- Created at Root Level, 18, 33

## Example

- C#, 14, 15
- Cache Hit, 10
- Cache Miss, 10
- Category Node, 22, 24, 37, 40
- Category Node Nested, 22, 25, 37, 40
- env\_def Override Default, 20, 35
- Heading Node, 20, 36
- Input Node, 25, 41
- Input Overrides, 26, 41
- Insbridge Rate Request Result XML Document, 22, 38
- Interface, 31
- Javascript, 14
- Multiple <rate> Nodes, 27, 44
- Multiple Line of Business Request, 29, 45
- Multiple Program Nodes, 27, 44
- Multiple Rate Request, 27, 44
- Multiple State Request, 28, 45
- Program Node, 24, 39
- Program Node w/Heading Option Request, 24, 39
- program\_id, 21, 36
- Rate Node, 20, 35
- renc Override Default, 35
- Result Node, 24, 39
- result\_def Override Default, 21, 36
- Single Rate Request, 27, 43
- Time Statistics, 26, 41
- XML Document, 19, 34

## Exceptions

- Cache, 11

**G**

## gen\_date

- Attributes, 23, 39

## Greater Than

- Invalid Characters, 35

**H**

## Heading Node

- Example, 20, 36
- Program Nodes, 20, 36

## HTTP

- Usage in SoftRater WSI, 12

## HTTP GET

- Supported Operations, 12

## HTTP POST, 13, 30

- Supported Operations, 12

## HTTP SOAP Proxy, 13, 30

**I**

## i

- Attribute, 22, 24, 25, 37, 38, 40

## IBSS

- Submit Insbridge XML, 32

## identifier

- Attribute, 22, 37

## Identifiers

- XML Arguments, 17, 33

## IIS

- SoftRater WSI, 12

## Input Data

- Marking Beginning, 21, 37

## Input Format

- InsbridgeRate.XML, 12

## Input Node

- Example, 25, 41

## Input Overrides

- Example, 26, 41

## Input Values

- Specifying, 25, 41

## InputMappingIdentifier

- Identifier, 17, 33

## InputMappingType

- Identifier, 17, 33

## Insbridge Framework Administrator

- Configuring Cache, 9
- Disabling Cache, 11
- Setting Up Defaults, 20, 35

## Insbridge Rate Request

- XML Document, 19, 34

## Insbridge Rate Request Result XML Document

- Example, 22, 38

## Insbridge.Fwk.APP.EAR

- Target JAR, 30

## InsbridgeRate.XML, 12

## Integration

- Software, 12
- Software, 30

## Interface

- Example, 31

- Interfaces
  - EJB, 30
  - URL to POST, 30
- Invalid Characters
  - Overriding Default, 20, 35
- Item level
  - Categories, 22, 24, 37, 40

## J

- Javascript Example, 14
- JNDI Path
  - Typical, 30, 31
- JSP Page
  - Web Service Interface, 30

## L

- Less Than
  - Invalid Characters, 35
- Line of Business
  - Attribute, 20, 35
- lob
  - Attributes**, 20, 23, 35, 39
- Lob
  - Identifier, 17, 33

## M

- Map Node
  - Children, 22, 25, 38, 40
- Mapped Variable
  - Storing Information, 9
- Mapped Variables
  - Cache, 9, 10
- Mapping Group
  - Overriding, 21, 36
- Microsoft Internet Information Server
  - SoftRater WSI, 12
- Microsoft SQL Server
  - SoftRater, 7
- Multiple <rate> Nodes, 29, 45
  - Example, 27, 44
- Multiple Line of Business Request
  - Example, 29, 45
- Multiple Program Nodes
  - Example, 27, 44
- Multiple Rate Request
  - Example, 27, 44
- Multiple State Request
  - Example, 28, 45

## N

- n
  - Attribute, 22, 25, 38, 40
- Node
  - Category, 21, 24, 37, 40
  - Heading, 20, 36
  - Input, 25, 40
  - Map, 22, 25, 37, 40

- Program, 21, 24, 36, 39
- Rate, 20, 35
- Result, 23, 39
- Running Time, 26, 42
- Start Time, 26, 42
- Stop Time, 26, 42

## O

- Options
  - MapRequest SOAP, 17, 33
  - Rating Integration, 16, 32
- Output Data
  - Marking Beginning, 24, 40
- Output Format
  - InsbridgeRate.XML, 12
- Output XML Document, 19, 34
- OutputErrorXPathLoc
  - Identifier, 18, 33
- OutputMappingIdentifier
  - Identifier, 17, 33
- OutputMappingStateful
  - Identifier, 17, 33
- OutputMappingType
  - Identifier, 17, 33
- OutputSchema
  - Identifier, 18, 33
- Overriding
  - Default Character Encoding, 20, 35
  - Default Mapping Group, 21, 36
  - Default Rating Environment, 20, 35

## P

- parent\_id
  - Attributes, 21, 36
- Ping
  - Web Services Interface, 12
- Policy Category
  - Children, 22, 24, 37, 40
- Process Message, 14
- ProcessAsyncMessage
  - Web Services Interface, 12
- ProcessCustomMessage
  - Web Services Interface, 12
- Processing
  - Data In and Out of System, 17, 33
  - SoftRater Engine, 12
- ProcessMessage
  - Web Services Interface, 12
- Program
  - Identifier, 17, 33
- Program Node
  - Example, 24, 39
- Program Node w/Heading Option Request
  - Example, 24, 39
- Program Nodes
  - In Header, 20, 36
- program\_id
  - Attributes, 21, 24, 36, 39
  - Example, 21, 36
  - Rules, 21, 36

- program\_ver
  - Attributes, 21, 36
- Proxy Classes
  - Generating, 13, 30
- Purging Misses
  - Cache, 10

## R

- Rate Node
  - Example, 20, 35
- Rate Request
  - Node to Mark Beginning, 20, 35
- Rate Request Document, 19, 34
- Rate Request Result
  - Marking Beginning, 23, 39
- RateManager Environment
  - Cache Exceptions, 11
- Rating
  - Multiple Lines of Business, 29, 45
  - Multiple States, 28, 45
- Rating Arguments, 16, 32
- Rating Environment
  - Overriding, 20, 35
- Rating Integration
  - Options, 16, 32
- Rating Request Response XML, 19, 34
- ReceiveAsyncMessage
  - Web Services Interface, 12
- renc, 20, 35
  - Attributes, 35
- Request XML
  - Result Nodes, 23, 39
- Requirements
  - Rate Node, 20, 35
- Result Documents
  - Setting Options, 16, 17, 32, 33
- Result Group
  - Enabled, 21, 37
- Result Node
  - Example, 24, 39
  - Output XML, 20, 35
- result\_def, 21, 36
- Root Level
  - Error Node Created, 18, 33
- Rules
  - Attribute, 21, 36

## S

- Sample
  - SOAP Proxy Class, 30
- Server
  - Restarted or Shutdown SoftRater, 11
- Setting
  - Target Hosting, 30
- Setting Options
  - Result Documents, 16, 17, 32, 33
- Several Rates
  - Requesting, 27, 44
- Single Quote
  - Invalid Characters, 35

- Single Rate Request
  - Example, 27, 43
- SOAP
  - Options, 17, 33
  - Supported Operations, 12
- SOAP Proxy Class
  - Sample, 30
- SoftRater
  - Definition, v
  - Engine Perform, 7
  - Input Overrides, 25, 41
- SoftRater Engine
  - Processing, 12
  - Rating Arguments, 16, 32
- SoftRater Explorer Configuration Settings
  - Updated, 11
- SoftRater Web Service, 13
- SoftRater WSI
  - Processing XML Data, 17, 33
- SoftRaterEJB Engine
  - Start Processing, 26, 42
  - Stop Processing, 26, 42
- SoftRaterEJB.jar
  - Target JAR, 30
- SoftRaterProxy Class, 30
- Software
  - Integration, 12, 30
- Status
  - Attributes, 24, 39
- Submit Insbridge XML
  - IBSS, 32
- Submitting
  - Rate Requests via XML, 19, 34
- Subscriber
  - Identifier, 17, 33
- Supported Operations, 12
- Switches
  - <program> node, 36

## T

- Target Hosting
  - Setting, 30
- Target JAR, 30
- Time Statistics, 26, 41
  - Example, 26, 41
- Tracking
  - Attributes, 20, 35

## V

- v
  - Attribute, 22, 25, 38, 40
- Variables
  - Storing Mapped Variable Information, 9
- ver
  - Attributes, 24, 39
- Version
  - Identifier, 17, 33
- version\_nm
  - Attributes, 24, 39

## W

- Web Service Interface, 12
- Web Service Interface ASP Page, 13
- Web Service Interface JSP Page, 30
- Web Services, 14
- WebSphere
  - EJB component, 30
- WSDL Documents, 13, 30
- WSI, 12

## X

- XML

- Insbridge, 19, 34
- Invalid Characters, 20, 35
- Usage in SoftRater, 19, 34
- Usage in SoftRater WSI, 12
- XML Data
  - Out of System, 16, 32
- XML Document
  - Example, 19, 34
  - Multiple Rate Requests, 27, 44
  - Returned, 26, 41
- XML Envelope, 24, 39