Oracle Insurance

Insbridge Enterprise Rating
SoftData Version 2
User Guide

For Java
Release 5.6

November 2018
CONTENTS

PREFACE .........................................................................................................................IV
Audience .........................................................................................................................v
Related Documents ........................................................................................................v
Conventions .....................................................................................................................v
System Requirements ....................................................................................................v
Manual History ..............................................................................................................v
SoftData Announcement ...............................................................................................vi

CHAPTER 1
INTRODUCTION TO SOFTDATA ..................................................................................7
Changes for SoftData Version 2 ..................................................................................7
Concept Diagram ..........................................................................................................8
Software Integration ......................................................................................................9
Arguments .....................................................................................................................10
Summary ......................................................................................................................11

CHAPTER 2
INSBRIDGE.XML DATA REQUEST FORMAT FOR 2.0 ....................................................12
Grouping .....................................................................................................................17
Default Value ..............................................................................................................18
OR Condition on Qualifiers .......................................................................................19
Sorting Results .............................................................................................................20
Global Versioning for the Program .............................................................................21
Environments ...............................................................................................................21

CHAPTER 3
INSBRIDGEDATA.XML RESULT FORMAT FOR 2.0 .....................................................22
Example .....................................................................................................................23
Summary .....................................................................................................................25

SUPPORT
CONTACTING SUPPORT ............................................................................................26

INDEX
INDEX .......................................................................................................................27
**List of Figures**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIGURE 1</td>
<td>LOCATION OF SOFTDATA IN IBSS</td>
<td>IV</td>
</tr>
<tr>
<td>FIGURE 2</td>
<td>LOCATION OF SOFTDATA WSDL IN IBSS</td>
<td>9</td>
</tr>
<tr>
<td>FIGURE 3</td>
<td>INSBRIDGEDATAREQUEST.XML TABLE</td>
<td>13</td>
</tr>
<tr>
<td>FIGURE 4</td>
<td>STANDARD CRITERIA</td>
<td>17</td>
</tr>
<tr>
<td>FIGURE 5</td>
<td>GROUPED CRITERIA</td>
<td>17</td>
</tr>
<tr>
<td>FIGURE 6</td>
<td>EXAMPLE INSBRIDGE DATA REQUEST.XML WITH GROUPING</td>
<td>18</td>
</tr>
<tr>
<td>FIGURE 7</td>
<td>EXAMPLE INSBRIDGE DATA REQUEST.XML WITH DEFAULT VALUE</td>
<td>18</td>
</tr>
<tr>
<td>FIGURE 8</td>
<td>EXAMPLE INSBRIDGE DATA REQUEST.XML WITH OR CONDITION</td>
<td>19</td>
</tr>
<tr>
<td>FIGURE 9</td>
<td>SORTING TABLE DATA</td>
<td></td>
</tr>
<tr>
<td>FIGURE 10</td>
<td>EXAMPLE INSBRIDGE DATA REQUEST.XML FOR A GLOBAL VERSIONING REPORT</td>
<td>20</td>
</tr>
<tr>
<td>FIGURE 11</td>
<td>INSBRIDGEDATARESULTS.XML</td>
<td>21</td>
</tr>
<tr>
<td>FIGURE 12</td>
<td>EXAMPLE INSBRIDGEDATARESULTS.XML</td>
<td>23</td>
</tr>
<tr>
<td>FIGURE 13</td>
<td>TABLE VARIABLE REPORT ON QUERIED TABLE</td>
<td>24</td>
</tr>
<tr>
<td>FIGURE 14</td>
<td></td>
<td>25</td>
</tr>
</tbody>
</table>
Preface

Welcome to the Oracle Insurance Insbridge Enterprise Rating SoftData Version 2 for Java Guide. This guide describes the usage and supported features of Oracle Insurance Insbridge Enterprise Rating SoftData Version 2 for Java (SoftData).

SoftData Version 2 is an updated version of SoftData Version 1 with expanded features. The updated SoftData Version 2 is still a feature of Oracle Insurance Insbridge Enterprise Rating SoftRater Server (IBSS) that allows for values to be retrieved dynamically from SoftRater Packages.

Soft Data is available from the IBSS Menu Tree.

![Figure 1 Location of SoftData in IBSS](image)

This guide serves as a supplemental document to the Insbridge SoftRater Server Guide. It provides a reference for developers to properly interact with the SoftData Engine either through SOAP or EJB Interfacing.
AUDIENCE

This guide is intended for system administrators who are tasked with administering SoftRater Server. A fundamental knowledge of SoftRater is required. Readers of this document should be familiar with XML, HTTP and the corresponding platforms; WebLogic, WebSphere or Jboss.

RELATED DOCUMENTS

For more information, refer to the following Oracle resources:

- The Oracle Insurance Insbridge Enterprise Rating RateManager User Guide.
- The Oracle Insurance Insbridge Enterprise Rating SoftRater 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:

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

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.
## Preface

<table>
<thead>
<tr>
<th>Edition</th>
<th>Publication Number</th>
<th>Product Version</th>
<th>Publication Date</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Edition</td>
<td>P01-774-01</td>
<td>R 5.1.1</td>
<td>March 2016</td>
<td>Initial</td>
</tr>
<tr>
<td>2nd Edition</td>
<td>P01-774-02</td>
<td>R 5.1.2</td>
<td>May 2016</td>
<td></td>
</tr>
<tr>
<td>3rd Edition</td>
<td>P01-774-03</td>
<td>R 5.2</td>
<td>July 2016</td>
<td></td>
</tr>
<tr>
<td>4th Edition</td>
<td>P01-774-04</td>
<td>R 5.4</td>
<td>January 2017</td>
<td>Update</td>
</tr>
<tr>
<td>5th Edition</td>
<td>P01-774-05</td>
<td>R 5.5</td>
<td>September 2017</td>
<td>Update</td>
</tr>
<tr>
<td>6th Edition</td>
<td>P01-774-06</td>
<td>R 5.6</td>
<td>August 2018</td>
<td></td>
</tr>
</tbody>
</table>

## SoftData Announcement

SoftData for Windows uses SoftData Version 1. SoftData Version 2 is a feature of Oracle Insurance Insbridge Enterprise Rating SoftRater Server (IBSS) that allows for values to be retrieved dynamically from SoftRater Packages and has expanded features such as, grouping and default values. SoftData Version 2 is only for Java environments only and will not be incorporated in to IBFA.

Moving forward enhancements may be added to SoftData Version 2 and may not be incorporated in SoftData Version 1. Bug fixes will still be addressed for SoftData Version 1.

**NOTE:** SoftData Version 1 can be used in IBSS for release 5.2 or greater, if the new features of grouping and default value are to be used, the entire request format must be changed to the format used by SoftData V2. The formats cannot be mixed.
INTRODUCTION TO SOFTDATA

SoftData is a method designed to provide the maximum amount of integration flexibility while maintaining a high level of operational efficiency for rules and rating applications. It allows an application to dynamically retrieve values from SoftRater Packages (SRPs) so that values do not have to be hard coded into an application.

For example, if you had a web-based application that allowed a consumer or agent to request a quote, SoftData calls could be used to fill in drop down text boxes with valid values. This allows the same application to be used across multiple states and carriers. It also prevents duplicate data entry, reducing the chance of making a costly mistake.

The core foundation is based on enumerated mappings to program based data elements built through the RateManager application. The schema model uses character mapping node items, i.e. <tv>, <q>, <i>, etc., which provide efficient document processing while gaining the system benefits of an extremely low XML document weight for small storage and faster transmission.

There are two parts to a SoftData call; the data request and the data result. For more information on these, see the following:

- Insbridge.XML Data Request Format for 2.0
- InsbridgeData.XML Result Format

Changes for SoftData Version 2

Backend changes to SoftData Version 2 now allow:

- The SoftData request to call the SoftRater engine to make the SoftData call.
- The SoftData tab on IBSS now has the option to return the request XML along with the results.
- SoftData Integration methods have been updated:
  - SoftData Version 1 will continue to use – GetMapData(requestXML,environment);
  - SoftData Version 2 uses – GetProgramVersionMapData(requestXML,environment,addRequestXML);

**NOTE:** SoftData Version 1 can be used in release 5.4.X or later, if the new features of grouping and default value are to be used, the entire request format must be changed to the format used by SoftData V2. The formats cannot be mixed.

The SoftData request is validated to recognize which version is being used.

An additional SoftData WebService method has been created for the new SoftData Version 2 request that uses the SoftRater engine to perform the following.
1. The SoftRater engine now can retrieve the program version that needs to be used by the SoftData request without using global versioning. A value added to the program_ver attribute determines the specific version to be used.

2. The SoftRater engine can retrieve the table variable revision number that the found program version is packaged to use. If you do not want to not use the table revision used in the program, an r (version number) attribute can be added to the <tv> element where you can specify the revision to use. The r attribute requires that the t (data type) attribute on the <q> element also in added.

3. A SoftData table query can be done using the name of the table. On the SoftData request, the d (table variable description) attribute must be included. The i (table variable identification number) attribute can still be used. If the i attribute and the d attribute are both included, the i attribute is used. If global versioning is not included to get the program version to use, the d attribute (table variable description) can be added to get the table used by the found program version.

4. The SoftRater engine can get the program ID of the table variable used in the selected program version.

5. When submitting a global versioning SoftData request, the following <tv> element attributes are required (i,r). When submitting a global versioning SoftData request, the following <q> element attributes are required (i,v,o)

6. Grouped criteria are arranged alphabetically or numerically in ascending order when the g (grouping) attribute on the <q> element has been added. The allowed values are 0 for no grouping or 1 to enable grouping.

**Concept Diagram**

The diagram below shows the high level interaction between the client application and the SoftData system. The SoftRater instance is responsible for all rating and underwriting processing but SoftData interface provides all the services for program data inquiry. The basic functionality of the SoftData system is demonstrated below. Input data is sent to the SoftData instance, processed and output data is returned.
SOFTWARE INTEGRATION

SoftData is an EJB component hosted in the Application Server and accessible through the following 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 SoftData Web Service WSDL can be found on the IBSS  Node Name.

![Figure 2 Location of SoftData WSDL in IBSS](image)

Whether sample SoftDataProxy 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. The target hosting URL may vary according to platform.

2. **EJB** – Direct JNDI interfacing.

   The EJB interfaces for creating service clients which are used to create SoftRater instances.

   Typical JNDI Path – This value is difference for each platform:
Target JAR – SoftRaterEJB.jar is a part of the SoftRater.EAR file* that you downloaded. It contains the following standard EJB 2.0 interface files.

- SoftData – Remote interface for Enterprise Bean
- SoftDataBean – Bean implementation class for Enterprise Bean
- SoftDataHome – Home interface for Enterprise Bean

*The SoftRater.EAR file depends upon your selected platform.

Interface Example:

```java
package com.insbridge.softdata;

import java.rmi.RemoteException;

/**
 * Remote interface for Enterprise Bean: SoftRater
 */
public interface SoftData extends javax.ejb.EJBObject
{
    public String GetProgramVersionMapData(
        final String XMLInput,
        final String TargetEnvironment,
        boolean aAddRequestXML
    )

    throws RemoteException;

    public void ResetConfigInfo() throws RemoteException;

    public String getDefaultPath() throws RemoteException;

    public String getVersion() throws RemoteException;

    public String QueryAvailableEnvironments() throws RemoteException;
}
```

Arguments

The SoftRater engine rating arguments control the handling of XML data out of the system. For optimal performance, use the following options, if possible, for your rating integration.

- String XMLInput – InsbridgeData.XML (Request) document containing the target data program data and lookup variables to be inquired against.
- String TargetEnvironment – (Optional) – Name of the SoftRater datasource environment to which the request should be processed against. If not provided the default will be obtained from the required target program data in the InsbridgeData.XML (Request).
- `aAddRequestXML` – adds the request XML to the results. True adds the request, false does not.

**SUMMARY**

- To request data from multiple program data sources you can include 1 – N number of program nodes in the `<datarequest>` node.
- To request data from multiple table variables you can include 1 – N mapping nodes in the `<program>` target node.
- All requests must have `project_id`, `parent_id`, and `program_id`.
- `env_def` can be left off. The environment selected from the drop down is used by default. If rating to any other environment an `env_def` is required. If rating using web services, an `env_def` is required.
- Operators for a global versioning report, if used, must in the same order as the program.
- The global versioning report requires the `product_id` on the `tv` node. An `r` element of zero (0) must also be included on the `tv` node. For example, `r="0"`
- Either the table variable identification number (i) or the variable name (ni) is required. When the table variable identification number (i) IS used, then the variable name (ni) is NOT required. When the table variable identification number (i) IS NOT used, then the variable name (ni) IS required.
- Scope is required for Globals and Universals. If the scope is not included in the request, only local table variables are queried.
- The program version number (program_ver) is not required. If the program version number is not supplied, global versioning is used to determine the program version to be used.
Chapter 2

**INSBRIDGE.XML Data Request Format for 2.0**

InsbridgeData.XML for SoftData has a slightly different format than previous SoftData versions. Since this format is run directly through SoftRater, the number of qualifiers is not required nor is a date mask. You can use either the table variable ID or the table variable name or both.

The program version can be left off. In the case that no global versioning is in place, the active version is used.

As with previous SoftData versions, more than one data request can be requested.

InsbridgeDataRequest.XML is the format in which requests are sent to SoftData. The table below outlines the nodes and attributes and whether or not they are required.

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>DATA TYPE</th>
<th>DESCRIPTION</th>
<th>REQUIRED</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ibdoc&gt;</td>
<td></td>
<td>Insbridge document namespace node. This is the requesting server, not the responding server.</td>
<td>Y</td>
</tr>
<tr>
<td>&lt;request&gt;</td>
<td></td>
<td>SoftRater namespace to indicate that request should be run through SoftRater.</td>
<td>Y</td>
</tr>
<tr>
<td>project_id</td>
<td>Long</td>
<td>The project identification number</td>
<td>Y</td>
</tr>
<tr>
<td>env_def</td>
<td>String</td>
<td>SoftRater Explorer environment identifier</td>
<td>N</td>
</tr>
<tr>
<td>&lt;heading&gt;</td>
<td></td>
<td>Heading target node.</td>
<td>Y</td>
</tr>
<tr>
<td>&lt;datarequest&gt;</td>
<td></td>
<td>Data request node</td>
<td>Y</td>
</tr>
<tr>
<td>&lt;program&gt;</td>
<td></td>
<td>Program target node</td>
<td>Y</td>
</tr>
<tr>
<td>parent_id</td>
<td>Long</td>
<td>Your company's subscriber ID</td>
<td>Y</td>
</tr>
<tr>
<td>program_id</td>
<td>Long</td>
<td>Program identification number</td>
<td>Y</td>
</tr>
<tr>
<td>program_ver</td>
<td>Long</td>
<td>Program version number. If not specified, global versioning is used.</td>
<td>N</td>
</tr>
<tr>
<td>&lt;tv&gt;</td>
<td></td>
<td>Table lookup variable node</td>
<td>Y</td>
</tr>
<tr>
<td>i</td>
<td>Long</td>
<td>Lookup variable identification number. *If no variable identification number is provided, a name must be provided.</td>
<td>N*</td>
</tr>
<tr>
<td>ni</td>
<td>Any</td>
<td>Lookup variable name. *If no variable name is provided, a variable identification number must be provided. If the name of the table is provided for a global or universal a scope does not need to be provided.</td>
<td>N*</td>
</tr>
<tr>
<td><strong>scope</strong></td>
<td>Long</td>
<td>By default, the system assumes that the table is local. This is required for global tables. If the value is left off, the table is local. If the table is global or universal, that attribute must be set to zero (0) scope=&quot;0&quot;</td>
<td>N</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---</td>
</tr>
</tbody>
</table>
| **empty_qual** | Long  | Returns empty fields. The default is 0.  
0=DO NOT return empty fields.  
1=Return empty fields  
This is an optional field. The number returned is the actual number of qualifiers in the table. | N |
| **sv** | String | Sort values either ascending or descending. Specify the value column that is to be used for sorting. Enter how the sort is to be done; Ascending = 0 and Descending = 1. The entries are separated by a vertical bar (|). For example sv="2 | 1" | N |
| **dv** | Long  | Default value. 1=returns a default value, 0=no default value. The default is for no default value. This option is helpful when you want to verify the rows returned. | N |
| **p** | Long  | Lookup variable data row position to start querying new data from | N |
| **c** | Long  | Lookup variable total count of data rows to be returned | N |
| **fq** | String | Returns a list of qualifiers filtered by the string value. Use commas to separate, (1,5,7) | N |
| **product_id** | Long  | The product identification number is only needed when using global versioning for the program. | N |
| **i** | Long  | For global versioning, a lookup variable identification number of zero (0) must be used. For example i="0" | N |
| **r** | Short | Lookup variable revision number. Used in global versioning only. Must be set to zero (0). For example, r="0"  
The attribute is not required in non-global versioning calls and is ignored if sent down. | N |
| **<q>** |  | Lookup variable qualification node | N |
| **i** | Long  | Qualifier query column number | Y |
| **v** | Any   | Qualifier query value | Y |
| **g** | Long  | Indicates a grouping is needed. 1=grouping is needed. This can be placed on any qualifiers of the same data type for regular table variables. If grouping has been added to a data source, this entry is not required. | N |
| **<or>** |  | Or Condition statement for qualifiers | N |
| **v** | Any   | The query value to be used. | N |

*Figure 3 InsbridgeDataRequest.XML Table*
Overriding Q Node Attributes

On the <q> node, two attributes can override the default values found on the table or data source.

<table>
<thead>
<tr>
<th>&lt;q&gt;</th>
<th>Lookup variable qualification node</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>Qualifier operation selection</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>For SoftData calls, the o attribute is not required. The operation selection found on the table can be overridden.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>For example, if there is a table where a qualifier is equal to 6 and now values that are 6 or greater need to be returned, a new operator can be used. Place an o attribute on the &lt;q&gt; node where o=&quot;&gt;=&quot; to override the equal selection found on the table. This returns values where the qualifier is greater than or equal to 6.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Only valid qualifier operators can be used.</td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>Indicates a grouping is needed. 1=grouping is needed.</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>If grouping has been added to a data source, the grouping can be overridden by changing the grouping to zero ( 0 ). This shuts off grouping.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Overrides can be added to data sources only.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>If grouping is not needed on a regular table variable, remove the g attribute on the &lt;q&gt; node.</td>
<td></td>
</tr>
</tbody>
</table>

- Valid Qualifier Operators are entered as follows.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equals</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&lt;:=</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>&gt;:=</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt;&gt;</td>
<td>Not equal to</td>
</tr>
</tbody>
</table>
Sample Request for Local Table Variable

Example of a local variable, using the ID number, being sorted by column number 4 in ascending order:

```xml
<ibdoc>
  <request project_id="3" env_def="IBSR_DB2">
    <heading>
      <datarequest>
        <program parent_id="8659" program_id="51" program_ver="2">
          <tv i="59" sv="4 | 1">
            <q i="4" v="CA" />
          </tv>
        </program>
      </datarequest>
    </heading>
  </request>
</ibdoc>
```

Sample Request for Global Table Variable

Example of a request by name of a linked global linked variable:

```xml
<ibdoc>
  <request project_id="1" env_def="IBSR_ORC">
    <heading>
      <datarequest>
        <program parent_id="8659" program_id="465" program_ver="1">
          <tv ni="BI_Limit_per_Occurrence" scope="0" />
          <tv ni="BI_Limit_Factor" scope="0">
            <q i="1" v="75287" />
            <q i="2" v="Dallas" />
          </tv>
        </program>
      </datarequest>
    </heading>
  </request>
</ibdoc>
```

Sample for Universal Table Variable

Example of a request of a universal variable with empty qualifiers being returned:

```xml
<ibdoc>
  <request project_id="3">
    <heading>
      <datarequest>
        <program parent_id="8659" program_id="32">
          <tv i="43" scope="0" empty_qual="1">
            <q i="2" v="75287" />
          </tv>
        </program>
      </datarequest>
    </heading>
  </request>
</ibdoc>
```
Sample Request for Local Table Variable Returning the Entire Table

Example of a local variable, using the ID number, being sorted by column number 4 in ascending order:

```
<ibdoc>
  <request project_id="3" env_def="IBSR_DB2">
    <heading>
      <datarequest>
        <program parent_id="8659" program_id="51" program_ver="2">
          <tv i="59" sv="4 | 1">
          </tv>
        </program>
      </datarequest>
    </heading>
  </request>
</ibdoc>
```

By removing the `<q>` node, the entire table will be returned.
Grouping

Grouping indicates that the qualifier fields are related and may be entered in any order. Grouping the qualifiers together reduces redundant table rows and makes tables easier to manage. Grouping allows users to perform SoftData table lookups similar to a SQL "AND" command.

For example, in a table with 5 criteria columns, the first three columns may be from the same value type/source, but not in a specific order. That is - the range values may be the same -- A and B and C -- but not in a specific order -- B and A and C, C and A and B, etc. … but the result is always the same, regardless of the order of the first three criteria values.

To access the specified columns as a group of data, reduce redundancy and more easily manage table data, grouping can be added. Grouping indicates that the fields can be entered in any order. In the lower example, the table data would effectively be reduced from 12 rows to 2 rows (group "A and B and C" and group "X and Y and Z").

<table>
<thead>
<tr>
<th>Criterion1</th>
<th>Criterion2</th>
<th>Criterion3</th>
<th>Criterion4</th>
<th>Criterion5</th>
<th>Result1</th>
<th>Result2</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>String</td>
<td>Integer</td>
<td>Integer</td>
<td>String</td>
<td>String</td>
<td>String</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>1</td>
<td>2</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>B</td>
<td>1</td>
<td>2</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>C</td>
<td>1</td>
<td>2</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>B</td>
<td>C</td>
<td>A</td>
<td>1</td>
<td>2</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>C</td>
<td>A</td>
<td>B</td>
<td>1</td>
<td>2</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>C</td>
<td>B</td>
<td>A</td>
<td>1</td>
<td>2</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>8</td>
<td>9</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>X</td>
<td>Z</td>
<td>Y</td>
<td>8</td>
<td>9</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Y</td>
<td>X</td>
<td>Z</td>
<td>8</td>
<td>9</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Y</td>
<td>Z</td>
<td>X</td>
<td>8</td>
<td>9</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Z</td>
<td>X</td>
<td>Y</td>
<td>8</td>
<td>9</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Z</td>
<td>Y</td>
<td>X</td>
<td>8</td>
<td>9</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Figure 4 Standard Criteria*

When grouping is applied, the table can be reduced.

<table>
<thead>
<tr>
<th>Criterion1</th>
<th>Criterion2</th>
<th>Criterion3</th>
<th>Criterion4</th>
<th>Criterion5</th>
<th>Result1</th>
<th>Result2</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>String</td>
<td>String</td>
<td>Integer</td>
<td>Integer</td>
<td>String</td>
<td>String</td>
</tr>
<tr>
<td>GROUP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>1</td>
<td>2</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td>8</td>
<td>9</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

*Figure 5 Grouped Criteria*

Grouped columns must be of the same data type. The row grouped columns values will be sorted in alpha-numeric order - numbers then alpha characters, ascending from left to right.

At runtime the engine will sort the criteria and then perform the table lookup.
When table data is brought into SoftData it is sorted. The sort order is defined by the data type. Data type string is alphabetized, data type decimal is put in numerical order, data type integer is put in numerical order and data type date is in ascending order. All string, decimal and integer data is also in ascending order. The grouping attribute indicates that the qualifiers assigned to group should be placed in order as well. If grouping is on the data source, then the grouping attribute can be used. 1 is on, 0 is off.

Grouping must be exclusive of criteria which are defined as "Interpolate" and/or "Wildcard".

Example of a request for a global variable with 5 criteria using grouping:

```xml
<ibdoc>
  <request project_id="5" env_def="RM">
    <heading>
      <datarequest>
        <program parent_id="8659" program_id="212" program_ver="1">
          <tv i="38" scope="0"/>
          <q i="1" v="Texas" g="1"/>
          <q i="2" v="California" g="1"/>
          <q i="3" v="1000"/>
          <q i="4" v="2017-01-01"/>
          <q i="5" v="New York" g="1"/>
        </program>
      </datarequest>
    </heading>
  </request>
</ibdoc>
```

*Figure 6 Example Insbridge Data Request.XML with Grouping*

**Default Value**

The default of the table is not stored in the table; it is stored in the runtime object of the program. In the event that you send down a request and no rows are returned, if default attribute is set to 1, the default value is returned. If the default value of the attribute is set to 0, no rows will contain the default value.

Example of a request for a local variable to return the default value:

```xml
<ibdoc>
  <request project_id="1" env_def="RM">
    <heading>
      <datarequest>
        <program parent_id="8659" program_id="91" program_ver="4">
          <tv i="10" ni="Criminal Count" dv="1"/>
          <q i="1" v="5"/>
        </program>
      </datarequest>
    </heading>
  </request>
</ibdoc>
```

*Figure 7 Example Insbridge Data Request.XML with Default Value*
OR Condition on Qualifiers

An OR condition can be placed on qualifiers. This allows for the qualifier to accept more than one value. For example, if you have a table variable where the value of the 5th criteria could be 1, 2 or 3, you could add an OR condition to the qualifier that would allow any one of the values to pass.

The values must be included in the <or v> attribute with an end </q>. This shows the value were v is one of the values listed. The operator is for all values.

Example of a local variable sorted by table value column 1, returning empty qualifiers and using the OR condition:

```xml
<ibdoc>
  <request project_id="2" env_def="rm">
    <heading>
      <datarequest>
        <program parent_id="8659" program_id="62" program_ver="2">
          <tv i="30" empty_qual="1" sv="1">
            <q i="1" v="ABC"/>
            <q i="2" v="12/31/2017"/>
            <q i="3" v="123"/>
            <q i="4" v="1.1"/>
            <q i="5" v="469">
              <or v="470"/>
              <or v="471"/>
            </q>
            <q i="6" v="YES"/>
          </tv>
          </program>
        </datarequest>
      </heading>
    </request>
  </ibdoc>
```

*Figure 8 Example Insbridge Data Request.XML with OR Condition*
Chapter 2 – InsbridgeData.XML Request Format

Sorting Results

In previous releases, SoftData results could not be sorted in ascending or descending order. This meant that when SoftData results populated outside application lists for example, the results may have been presented to users in a way that forced them to scroll for the option they wanted. In addition to being inconvenient, it could slow down application completion time and is counter intuitive to how most users select options.

Sorting allows for SoftData results to be returned in either ascending or descending order by the table variable value that you select. For example, if you have linked table variable with BI, PD, Comp, and Coll, columns, you can have SoftData results returned by any one of those column values in ascending or descending order.

Sorting cannot be done on criteria.

![Figure 9 Sorting Table Data](image)

To sort, an sv attribute must be added to the m node. The sv attribute is where you specify the column that is to be used for sorting.

Format for the sv attribute is: column # | ascending or descending

- Column # is the column as presented on the Edit Data screen. In Figure 3, if sorting was to be done by PD Territory Factor, a column number of 2 would be used.

- How the sort is to be done. Ascending = 0 and Descending = 1 The default is ascending. This value can be left off if the sort is ascending.

The entries are separated by a vertical bar ( | ).

For example: sv="2 | 1"

This would sort the listing by column 2, PD Territory Factor in descending order.

Sort and paging are mutually exclusive. Sorting can only be done on one column.

The sorting attribute should be used alone. There may be conflicts with other attributes.

**SoftData Version 2 Sorting Example**

Example of sorting by column number 2, PD Territory Factor using SoftData format 5.0.

```xml
<ibdoc>
  <request project_id="3" env_def="RM">
    <heading>
```
<program parent_id="8659" program_id="623458646" program_ver="7">
  <tv i="52" scope="0" sv="2 | 0">
    <q i="1" v="4"/>
  </tv>
</program>
</datarequest>
</heading>
</request>
</ibdoc>

Global Versioning for the Program

To obtain the global versioning for the program, the program node requires that the id be set to the program XML id found in RateManager and version attribute be set to zero (0). On the table lookup node, the identification node must be set to zero (0) and the product ID must be included.

Operators for global versioning report must in the same order as the program. For example, if your program uses the effective date to determine what program version should be run and the date values are listed as less than (q1) and greater than (q2). You need to list those values in the request as less than first and greater than second.

<ibdoc>
<request project_id="3" env_def="RM">
  <heading>
    <program parent_id="8659" program_id="62" program_ver="1">
      <tv i="0" r="0" product_id="1"/>
    </tv>
  </program>
  <datarequest>
    <program>
      <datarequest>
        </program>
        </datarequest>
      </heading>
      </request>
    </ibdoc>

Figure 10 Example Insbridge Data Request.XML For a Global Versioning Report

Environments

If submitting via web services, an environment must be specified in the request.

If using the IBSS SoftData tester, the environment must be selected from the drop down node option. An environment can be specified in the request. If an environment is specified, the environment selected in the dropdown is not used. If no environment is selected an error message is presented.
InsbridgeDataResults.XML is the format that results are received in from a data request. The table below shows the information returned.

<table>
<thead>
<tr>
<th>ELEMENT</th>
<th>DATA TYPE</th>
<th>DESCRIPTION</th>
<th>ADDITIONAL INFORMATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;ibdoc&gt;</td>
<td></td>
<td>Insbridge document namespace node. This is the requesting server, not the responding server.</td>
<td></td>
</tr>
<tr>
<td>&lt;result&gt;</td>
<td></td>
<td>Result node</td>
<td></td>
</tr>
<tr>
<td>project_id</td>
<td>Long</td>
<td>The project identification number</td>
<td></td>
</tr>
<tr>
<td>env_def</td>
<td>String</td>
<td>SoftRater Explorer Environment Identifier</td>
<td></td>
</tr>
<tr>
<td>gen_date</td>
<td>Datetime</td>
<td>Document creation time stamp</td>
<td></td>
</tr>
<tr>
<td>ibdoc_version</td>
<td>Long</td>
<td>The ibdoc version used</td>
<td></td>
</tr>
<tr>
<td>engine_type</td>
<td>String</td>
<td>The application server type used</td>
<td></td>
</tr>
<tr>
<td>site_location</td>
<td>String</td>
<td>The name of the physical server</td>
<td></td>
</tr>
<tr>
<td>site_instance</td>
<td>String</td>
<td>The name of the instance used</td>
<td></td>
</tr>
<tr>
<td>&lt;dataresult&gt;</td>
<td></td>
<td>Data result node</td>
<td></td>
</tr>
<tr>
<td>&lt;program&gt;</td>
<td></td>
<td>Program selected node</td>
<td></td>
</tr>
<tr>
<td>parent_id</td>
<td>Long</td>
<td>Your company’s subscriber ID</td>
<td></td>
</tr>
<tr>
<td>program_id</td>
<td>Long</td>
<td>Program identification number</td>
<td></td>
</tr>
<tr>
<td>program_ver</td>
<td>Long</td>
<td>Program version number</td>
<td></td>
</tr>
<tr>
<td>package_date</td>
<td>Datetime</td>
<td>Date of package creation</td>
<td></td>
</tr>
<tr>
<td>region_format</td>
<td>Long</td>
<td>The local used</td>
<td></td>
</tr>
<tr>
<td>status</td>
<td>Long</td>
<td>Pass or Fail status</td>
<td></td>
</tr>
<tr>
<td>from_cache</td>
<td>Long</td>
<td>Package in cache</td>
<td></td>
</tr>
<tr>
<td>&lt;tv&gt;</td>
<td></td>
<td>Table lookup variable node</td>
<td>One node is returned for each corresponding node in the data request</td>
</tr>
<tr>
<td>i</td>
<td>Long</td>
<td>Lookup variable identification</td>
<td></td>
</tr>
<tr>
<td>r</td>
<td>Short</td>
<td>Lookup variable identification revision number</td>
<td></td>
</tr>
<tr>
<td>product_id</td>
<td>Long</td>
<td>The product identification number</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
<td>-----------------------------------</td>
<td></td>
</tr>
<tr>
<td>p</td>
<td>Long</td>
<td>Last data item row position retrieved</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>Long</td>
<td>Lookup variable total count of data nodes returned</td>
<td></td>
</tr>
<tr>
<td>ni</td>
<td>Any</td>
<td>Lookup variable name from the input</td>
<td></td>
</tr>
<tr>
<td>l</td>
<td>Boolean</td>
<td>Lookup variable flag indicating if the result contains linked results</td>
<td></td>
</tr>
<tr>
<td>status</td>
<td>Long</td>
<td>Pass or Fail status</td>
<td></td>
</tr>
</tbody>
</table>

| <d>        | Data node | One node is returned for each row returned |
| p          | Long      | Data row position indicator |

| <v>        | Any       | Value node (Multiple are returned for linked table variables) | One node is returned for each variable |

| <q>        | Any       | Lookup variable qualification node | One node is returned for each qualifier |

| <stats>    | Data result node |
| start_time | Datetime  | The start time |
| stop_time  | Datetime  | The stop time |
| running_time | Datetime | The run time |
| xml_walking | Datetime  | The time span it takes to process the XML file. The engine has to process the file from top to bottom before the rating process can start. |

Figure 11 InsbridgeDataResults.XML

An example data result is shown below.

Example

```xml
<ibdoc>
  <result project_id="1" env_def=" SR_win" gen_date="2016-06-16 1:25:28 PM"
  Ibdoc_version="3.1" engine_type="weblogic" site_location="IBSS_WL" site_instance="DEV">
    <dataresult>
      <program parent_id="8659" program_id="35" program_ver="2" package_date="2016-04-10T13:02:00 region_format="en-US" status="PASS" from_cache="true">
        <tv i="10" r="2" ni="Bl_limit_per_Occurrence" l="true" status="PASS">
          <d p="1">
```
<v>100/200</v>  
<v>Our_Standard_Limit</v>  
<v>L100</v>  
<v>100</v> 
</d> 
<d p="2">  
<v>300/400</v>  
<v>Optional_Limit</v>  
<v>L200</v>  
<v>200</v> 
</d> 
<d p="3">  
<v>200/300</v>  
<v>Highest_Limit</v>  
<v>L300</v>  
<v>300</v> 
</d> 
</tv> 
<tv i="1" r="2" ni="BI_Limit_Factor" l="true" status="PASS">  
<d p="1">  
<v>0.001</v>  
<v>75025</v>  
<v>Plano</v>  
<v>Collin</v>  
<v>Texas</v> 
</d> 
<d p="2">  
<v>0.235</v>  
<v>75025</v>  
<v>Plano</v>  
<v>Collin</v>  
<v>Texas</v> 
</d> 
<d p="3">  
<v>0.906</v>  
<v>75025</v>  
<v>Plano</v>  
<v>Collin</v>  
<v>Texas</v> 
</d> 
</tv> 
</program> 
</dataresult> 
<stats>  
<start_time>2016-06-18 06:08:26:0166 PM</start_time>  
<stop_time>2016-06-18 06:08:26:0213 PM</stop_time>  
<running_time>47</running_time>  
<xml_walking>47</xml_walking> 
</stats> 
</result> 
</ibdoc> 

*Figure 12 Example InsbridgeDataResults.XML*
SUMMARY

- For each target program node there is one selected program node supplied in the <dataresults> node. Each <program> contains all queried table variables and data for that program.

- Table lookup variable nodes with the linked variable flag l=true contain 1 – N value nodes <v> for each data <d> row node returned. The values in the <v> nodes are assigned respective to the order determined during variable setup in (RateManager – Linked/Table Variables) and listed in the Inbridge Published Program Summary Report.
Support

CONTACTING SUPPORT

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

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

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
Index

INDEX

<
  <d>
    Requirements Table, 23
  <datarequest>
    Requirements Table, 12
  <dataresult>
    Requirements Table, 22, 23
  <dataresult> Node, 25
  <ibdoc>
    Requirements Table, 12, 22
  <m>
    Requirements Table, 12, 22
  <program>
    Requirements Table, 12, 22
  <program> Node, 25
  <q>
    Requirements Table, 13, 14, 23
  <v>
    Requirements Table, 23

A
  Attribute
    Requirements Table, 12

D
  Data Request, 7
  Data Results, 7
  Default
    Environment, 21

E
  Edition Notice, 2
  Environments, 21
  Example
    Global Versioning, 21
    Local Request, 18
    Result, 23
    SoftData, 7

F
  Format
    Results Recieved from Data Request, 22

G
  Global Versioning, 21

I
  Information Returned
    Requirements Table, 22
  InsbridgeDataResult.XML, 22

L
  Local Data Request, 12

N
  Nodes
    Requirements Table, 12

O
  Overview
    SoftData, 7

P
  Program Node
    Target, 25

Q
  Qualifier Operators
    Valid, 14
  Queried Table Variables, 25

R
  Request
    Local, 12
  Requesting Data
    Multiple Program Data Sources, 11
    Multiple Table Variables, 11
  Requirements Table
    Results Returned, 22

S
  SoftData
    Example, 7
    Overview, 7
  SoftData Definition, iv
  Support, 26

T
  Table Lookup Variable Nodes, 25
  Table Variable Node Items, 7