ORACLE
INSURANCE

Oracle Insurance

Insbridge Enterprise Rating
ESI for Java User Guide

Release 5.1

December 2015

Oracle Insurance Insbridge Enterprise Rating ESI for Java User Guide

Release 5.1.x

Part # E69577-01

Library # E69590-01

December 2015

Primary Authors: Aidi Nui, Mary Elizabeth Wiger

CONTENTS

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 12

SUPPORT

INDEX

LIST OF FIGURES

PREFACE

Welcome to the *Oracle Insurance Insbridge Enterprise Rating ESI for Java Guide*. This guide describes how you can configure and use the Insbridge Extended Services Interface (ESI).

AUDIENCE

This guide is intended for application developers, programmers and others tasked with configuring the ESI interface. Users should be familiar with Java IDE.

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 Javadoc File.
- The Oracle Insurance Insbridge Enterprise Rating Framework Administrator User Guide.
- You can view these guides in-line at this address:

  http://www.oracle.com/technology/documentation/insurance.html

CONVENTIONS

The following text conventions are used in this document:

Convention	Description
bold	Boldface type indicates graphical user interface elements associated with an action.
italic	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.

Oracle Insurance recommends the following system requirements for client workstations for optimal performance. Currently, Insbridge ESI only supports US English.

Insbridge ESI Clients:

- Operating System – Windows 7 or later
- JDK 6
- Unzip utility for using ESI system configuration

Manual History

New editions incorporate any updates issued since the previous edition.

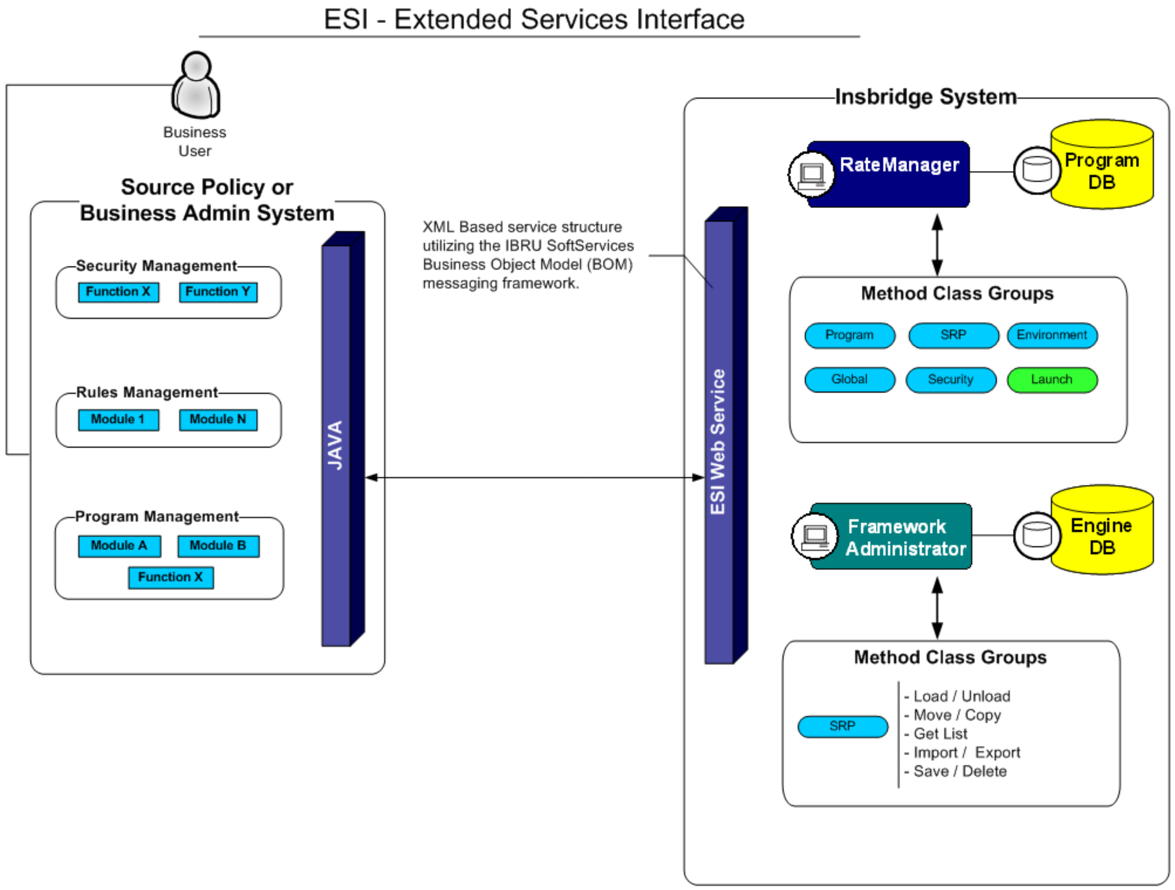
Edition	Publication Number	Product Version	Publication Date	Comment
1st Edition	P01-783-01	R 4.1	December 2010	Initial release
2nd Edition	P01-783-02	R 4.5	May 2011	Update
3rd Edition	P01-783-03	R 4.5.1	September 2011	Update
4th Edition	P01-783-04	R 4.6	May 2012	Update
5th Edition	P01-783-05	R 4.6.1	November 2012	Update Release
6th Edition	P01-783-06	R 4.7	September 2013	Update Release
7th Edition	P01-783-07	R 4.7.1	November 2013	Update
8th Edition	P01-783-08	R 4.8	August 2014	Update
9th Edition	P01-783-09	R 4.9	December 2014	Update
10th Edition	P01-783-10	R 5.1	December 2015	Update

Chapter 1

INTRODUCTION TO INSBRIDGE EXTENDED SERVICES INTERFACE

Oracle Insurance Insbridge Enterprise Rating Extended Services Interface (ESI) is a library module designed to provide remote services to the Insbridge Enterprise Rating (Insbridge) business services and Insbridge Soft Service (IBSS) without directly utilizing the system User Interfaces (UI).

ESI for Insbridge



ESI allows users to pass information between a user's source policy or business admin system and the RateManager system or the Insbridge Framework Administrator (IBFA) without using either the RateManager or IBFA UI. ESI itself is not a web service. You reference it in your applications and **consume it natively**. ESI then calls the web services in RateManager or IBFA. Both RateManager and IBFA have a web service interface that ESI can call.

ESI is designed to operate on Insbridge version 04.01.00 or higher and connects with:

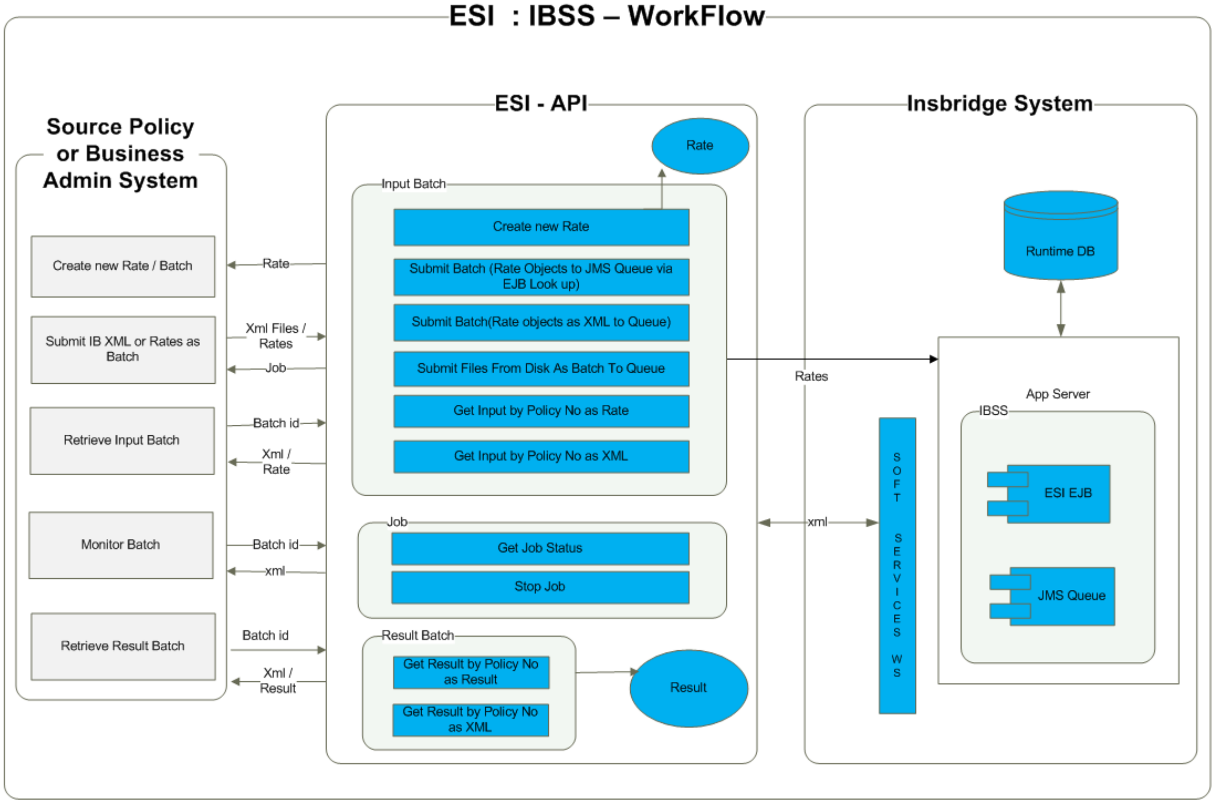
- Oracle Insurance RateManager Interface
- Oracle Insurance Framework Administrator Interface

For example, if you had a vendor Policy Administration system and wanted to provide integrated product management through your custom solution; ESI would allow you to model and use the workflow on the Insbridge business services and functions but control all the user presentation aspects through your custom product interface.

Rule and Functionality Highlights

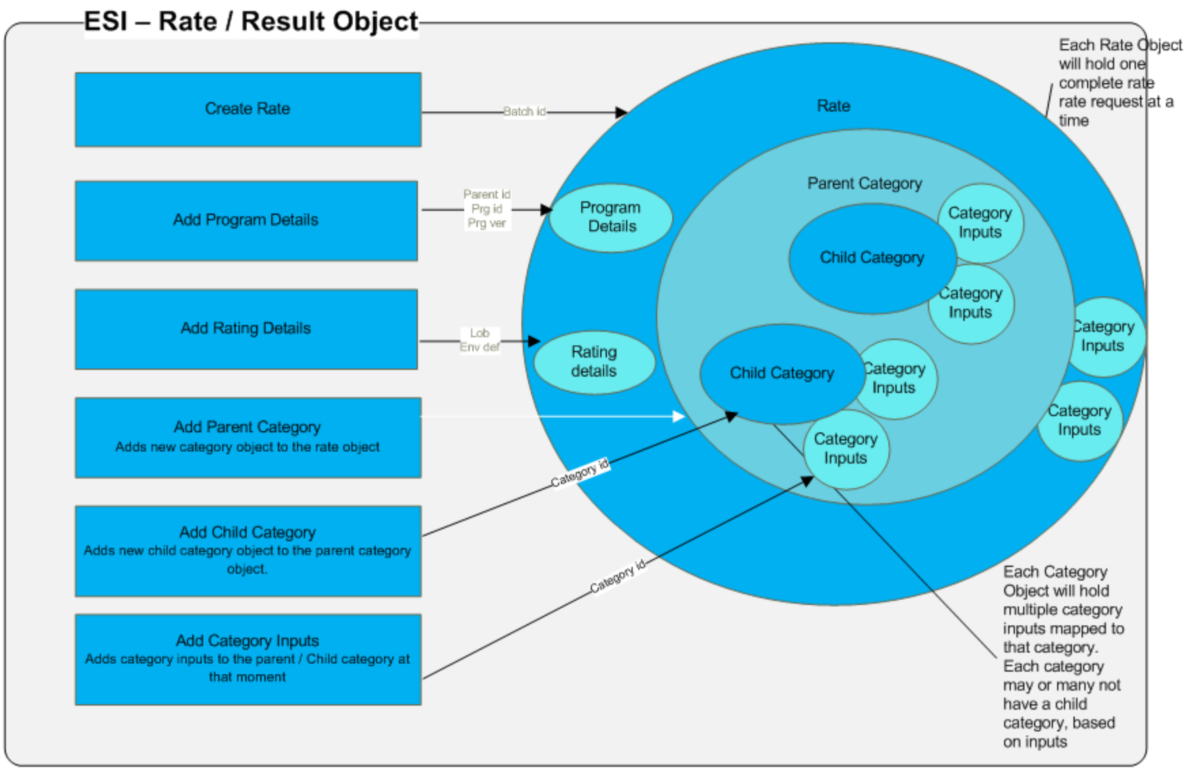
- ESI is available on Insbridge Enterprise Rating version 04.01.00 or higher.
- ESI is shipped with the IBFA installation. To locate ESI, look under the Insbridge directory for the SDK/ESI/Java file; [INSTALL_DIR]\Insbridge\SDK\ESI\Java. This file contains five items:
  - InsbridgeEsiServices.jar
  - EsiSwingTester.cmd
  - EsiSwingTester.jar
  - EsiSwingTester Source Code.zip
  - Javadoc.zip
- ESI entries are immediate. There is no lag time or system sync required.
- An ESI cannot be recalled or undone. If an ESI needs to be edited, for example a folder has an incorrect name; you must enter RateManager or IBFA and make the changes there.
- ESI works with the default subline in the folder of your choice for any line of business.
- Unless specifically stated as Optional, all Inputs are **Required**.

ESI Work Flow for IBSS



NOTE: *For further details, please see the Javadoc file.*

Rate Model ESI: IBSS



Each Rate Object will hold one complete rate rate request at a time

Each Category Object will hold multiple category inputs mapped to that category. Each category may or many not have a child category, based on inputs

Functionality Highlights

- A rate request now can be an Object (Rate) with rate request details or an Insbridge XML.
- With the ESI API, you can submit an InputBatch as a collection of rate objects or Insbridge XML, to the runtime database of IBSS.
- Using the InputBatch, input rates can be retrieved either as an XML or as a rate object.
- Using the Result Batch, the result of a rate can be retrieved either as an XML or as a result object.
- When an InputBatch is submitted for execution, a Job is returned that can be used to get the status of the Job, or stop the Job even if it is being executed.
- The status can be retrieved either as an XML or as a status object.

Chapter 2

Using ESI: IBSS - Example

Creating a Rate Object

ESI has a Rate Object that can be used to create a rate request. A Rate Object is the representation of a rate request. A collection of rate requests is an InputBatch that can be submitted to the runtime database for execution as a batch.

IBRequestResponse Interface

Rate implements the IBRequestResponse interface. You can use the interface to create rate objects to add to the runtime database as a batch for execution.

```
public interface IBRequestResponse {
   public void addRatingDetails(String lob, String policyNumber, String env_def);
   public void addProgramDetails(String parent_id, String tiering_id, String program_id, String
   program_ver);
   public int addParentCategory(String catID, String desc);
   public int addChildCategory(String catID, String desc);
   public void addCategoryInputs(String id, String name, String value);
   public boolean clear() ;
   public int getParentCategorySize();
   public HashMap getParentCategoryMap();
   public void setCurrentCategory(CategoryVO currentParent);
   public CategoryVO getCurrentCategory();
   public void setRateDetails(RateVO rateDetails);
   public RateVO getRateDetails();
   public void setProgramDetails(ProgramVO programDetails);
   public ProgramVO getProgramDetails();
}
```

Creating a Rate Object

Step 1: Create an InputBatch.

Step 2: Use the createNewRate() method to create a new rate request object.

Step 3: Use the addRatingDetails() method to add rating details to the rate.

Step 4: Use the addProgramDetails() method to add program details to the rate.

Step 5: Use the addParentCategory method to add a parent category to the rate.

A parent category is always at the top level. Child categories can be added underneath the parent category.

Step 6: Use the addCategoryInputs() method to add category inputs to the category created in Step 5.

The addCategoryInputs() method adds category inputs to the current category defined in Step 5.

Step 7: Use the addChildCategory() method to add a child category to the parentCategory or current category.

Step 8: Use the addCategoryInputs() method to add category inputs to the category.

Step 9: Use the setCurrentCategory() method to set the current category where you want to work adding a child category or category inputs.

Example

```
rateRequest.setCurrentCategory((rateRequest.getCurrentCategory().getParentCategory()).getParentCategory());
```

Sample Code for Creating a Rate Object

```
public class SampleRateObject {
   public static void main(String[] args) {

         InputBatch inputBatch = new InputBatch();
         Rate rateRequest = inputBatch.createNewRate();

         //create rate and program objects for the batch

         rateRequest.addRatingDetails("111", "", "rm");
         rateRequest.addProgramDetails("775", "", "2", "1");

         //Start creating new category objects by adding new Parent category

         rateRequest.addParentCategory("12", "vehicle");

         //add category inputs for the Parent category

         rateRequest.addCategoryInputs("20", "CompanyCd", "Alamere");
         rateRequest.addCategoryInputs("21", "UseLowerFalseClaimFactorInd", "");
         rateRequest.addCategoryInputs("24", "GroupCd", "");
         rateRequest.addCategoryInputs("46", "EffectiveDt", "03/17/2015");

         // adding child category

         rateRequest.addChildCategory("12_1", "vehicle_1_CAR");
         rateRequest.addCategoryInputs("12_1", "CompanyCd", "gershome");
         rateRequest.addCategoryInputs("12_1", "GroupCd", "gershome");
         rateRequest.addCategoryInputs("12_1", "EffectiveDt", "03/17/2015");
         rateRequest.addCategoryInputs("12_1", "UseLowerFalseClaimFactorInd", "");
```

```
          rateRequest.addChildCategory("12_1_1", "vehicle_1_CAR_1");
          rateRequest.addCategoryInputs("12_1_1", "CompanyCd", "gershome");
          rateRequest.addCategoryInputs("12_1_1", "GroupCd", "gershome");
          rateRequest.addCategoryInputs("12_1_1", "EffectiveDt", "03/17/2015");
          rateRequest.addCategoryInputs("12_1_1", "UseLowerFalseClaimFactorInd", "");

           // How to change from one Parent Category to another

          rateRequest.setCurrentCategory(rateRequest.getCurrentCategory().getParentCategory());

          rateRequest.addChildCategory("12_1_2", "vehicle_1_1CAR_2");
          rateRequest.addCategoryInputs("12_1_2", "CompanyCd", "gershome");
          rateRequest.addCategoryInputs("12_1_2", "GroupCd", "gershome");
          rateRequest.addCategoryInputs("12_1_2", "EffectiveDt", "03/17/2015");
          rateRequest.addCategoryInputs("12_1_2", "UseLowerFalseClaimFactorInd", "");

          // Sample setting up parent category for new Child categories at different levels

rateRequest.setCurrentCategory((rateRequest.getCurrentCategory().getParentCategory()).getParentCate
gory());
          rateRequest.addChildCategory("12_2", "vehicle_1_bus");
          rateRequest.addCategoryInputs("12_2", "CompanyCd", "gershome1");
          rateRequest.addCategoryInputs("12_2", "CompanyCd", "gershome2");

          // adds a parent category at top level

          rateRequest.addParentCategory("13", "Driver");
          rateRequest.addCategoryInputs("20", "CompanyCd", "Alamere");
          rateRequest.addCategoryInputs("21", "UseLowerFalseClaimFactorInd", "");
          rateRequest.addCategoryInputs("24", "GroupCd", "");
          rateRequest.addCategoryInputs("46", "EffectiveDt", "03/17/2015");
} }
```

NOTE: *Once moved up from a child category to its parent category, system cannot go down to the child category.*

Input Batch – Creating, Submitting, Executing and Monitoring

Creating an InputBatch

- An input batch is created using InputBatch.
- Create rates for the InputBatch using the createNewRate() method.
- Rates created with the createNewRate method are added to the InputBatch using the addRateToBatch() method.

Submitting an InputBatch

- Once all the rates have been added to the InputBatch, the InputBatch is added to the runtime database using the addInputBatchToRuntimeDB() method.

Executing an InputBatch

- An InputBatch in the runtime database can be executed to rate against the engine from ESI using the executeBatch() method.
- When an InputBatch is executed, a Job is returned to the user.

Monitoring a Batch

When the Job is returned from the executing of the InputBatch, you can monitor the batch.

Monitoring includes:

- Get the status of the InputBatch
  - getJobStatusAsXML to get the status as XML
  - getJobStatus() to get the status as status object (StatusVO)
- Stop the InputBatch (Job) using stopJob() method, even if the Job is currently being executed.

Retrieving Inputs

Input rates can be retrieved from the runtime database in two ways.

- Retrieve input rate by policy number as a rate object using the getInputRateByPolicyNumber() method.
- Retrieve input rate by policy number as XML using the getInputXMLByPolicyNumber() method.

Result Batch

Retrieving Results

Result can be retrieved from the runtime database in two ways.

- Retrieve result by policy number as a result object using the getResultByPolicyNumber () method.
- Retrieve result by policy number as XML using the getResultXMLByPolicyNumber () method.
- Once results have been retrieved, they can be added to the ResultBatch using the addResultToResultBatch() method.

Chapter 3

BASE DOMAIN TYPES

Base ESI domain types represent the business nouns within the Insbridge system. These data structures represent the majority of the business payload.

Please see the Javadoc file for details.

USE CASE SUMMARY

The Use Case Summary table is a quick guide to the function available for ESI for Insbridge.

Business Use Case	Methods to Use	Result
General – Open URL		
Create Launch Definition – using a Network login	General – createLauchDefinition (Selected Version, Network Login)	Method returns a URL for a selected program version using a network login.
Create Launch Definition – using a Network login	General – createLauchDefinition (First Version, Network Login)	Method returns a URL for the first program version using a network login.
Create Launch Definition – using a RateManager login	General – createLauchDefinition (Selected Version, RateManager Login)	Method returns a URL for a selected program version using a RateManager login.
Create Launch Definition – using a RateManager login	General – createLauchDefinition (First Version, RateManager Login)	Method returns a URL for the first program version using a RateManager login.
Programs		
Add	Program Items – createProgram	Create a new program.
Add New Version	Program Items – createProgramVersion	Creates a new version of an existing program.
Copy a Program	Program Items – copyProgram	Copies an existing Insbridge program.
Delete	Program Items – deleteProgram	Removes a program.
Update	Program Items – updateProgram	Updates a program.
Update	Program Items – updateProgramVersionLock	Updates a program version lock.

Folder Groups		
Add	Program Items – createFolder	Creates a new folder.
Delete	Program Items – deleteFolder	Removes a folder.
List Folders	Program Items – getFoldersXML	Returns an XML string of all available folders in an LOB.
Update	Program Items – updateFolder	Updates the name or location of a folder.
Line of Business		
Add	Program Items – createLine	Creates a new Line.
Delete	Program Items – deleteLine	Removes an Line.
Update	Program Items – updateLine	Updates an Line.
SRP – RateManager		
Add	SRP Items – createSRP	Creates an SRP and returns the SRP object.
Global Items		
Add	Global Items – createInput	Creates a new input.
Add	Global Items – createCategory	Creates a new category.
Add	Global Items – createCategory with Inputs	Creates a new category with inputs.
Delete	Global Items – deleteInput	Removes an input.
Delete	Global Items – deleteCategory	Removes a category.
Updates	Global Items – updateInput	Updates a category.
Updates	Global Items – updateCategory	Updates an input.
List Inputs	Global Items – getInputsArray	Returns an array of all inputs in a Line.
List Inputs	Global Items – getCategoryXML	Returns an XML string of all categories in a Line.
Security Items		
Add	Security Items – createUser	Creates a new user.
List	Security Items – getUsers	Returns a list of all avaialable users for the subscriber.
List	Security Items – getUserGroups	Returns a list of all avaialable user groups.
Login	Security Items – login	Logs in the user.

Login	Security Items – networkLogin	Logs in the user with network login.
Logout	Security Items – logout	Logs out the user.
Update	Security Items – updateUser	Updates a user’s information.
Environment Items		
List	Environment Items – getAvailableEnvironments, All Non-Secured	Returns a list of all available non-secured environments.
List	Environment Items – getAvailableEnvironments, Secured	Returns a list of all available environments within a selected environment type.
Releases		
Add	Releases Items – createRelease	Creates a new release.
Delete	Releases Items – deleteRelease	Removes a release.
List	Releases Items – getReleasePrograms	Returns a list of all programs in release.
Update	Releases Items – addReleaseProgram	Add a program version to a release.
Delete	Releases Items – removeReleaseProgram	Removes a program from the release.
SRP– IBFA		
Copy	SRP Items – copySRP	Copies an SRP from the current environment into a selected environment.
Delete	SRP Items – deleteSRP	Deletes an SRP from the disk.
Export	SRP Items – exportSRP	Exports an SRP from IBFA and converts it to bytes.
Import	SRP Items – importSRP	Imports SRP bytes into an IBFA environment.
List	SRP Items – getSRPList	Returns a list of all SRPs in a select environment.
List	SRP Items – getBytesFromFile	Returns the bytes from a file.
Load	SRP Items – loadSRP	Loads an SRP into an IBFA environment.
Move	SRP Items – moveSRP	Moves an SRP from the current environment to the passed environment.
Save	SRP Items – saveFile	Saves the bytes you pass into a physical file.
Unload	SRP Items – unloadSRP	Unloads an SRP from a selected environment.

The new ESI features added in 5.1.0 are:

Add/update program version name

Add/update program versioning data (Delete, Update, Insert with multiple rows)

Add/update notes to a program version (Retrieve all, Add, Update, Delete)

Add/update notes to a table revision (Retrieve all, Add, Update, Delete)

Define a new table revision

Export tables in the tab-delimited tab format

QUICK GUIDE TABLE

The quick guide table shows the functions available for ESI for IBSS.

Business Use Case	Methods to Use	Result
Input Batch		
Create new rate objects	createNewRate	Creates a new rate object that can be added to an InputBatch.
Add batch to DB	addInputBatchToRuntimeDB	Once all the rates have been added to the InputBatch, the batch is added to the runtime DB.
Execute the batch submitted to Runtime DB	ExecuteBatch	After an InputBatch is added to the runtime DB, it is executed (rated) against the engine.
Retrieve input rate	getInputRateByPolicyNumber	The input rate request is retrieved from the runtime DB either as an XML or as a rate object.
Submit Insbridge XML files from disk as batch	submitIBFilesFromDiskAsBatchToQueue	A collection of XML rate files in the disk is submitted as an InputBatch.
Submit Insbridge XML from memory	submitIBXMLFromMemoryToDB	A single XML rate from memory is submitted to the runtime DB as a batch or added to a batch.
Submit single rate object to database	submitRateObjectToDB	A single rate object is submitted to the runtime DB as a batch.
Result Batch		
Create new result object	createNewResult	Creates a new result object.
Get result by policy number	getResultByPolicyNumber	Returns a result either as an XML or a result object.
Job		
Get the status of the job	getJobStatus	Returns the status of the job submitted for execution.
Stop the job	stopJob	Stops the submitted job.

Rate / Result		
Add rating details to rate	addRatingDetails	Adds rating details to the rate.
Add program details to rate	addProgramDetails	Adds program details to the rate.
Add parent category to rate	addParentCategory	Adds parent category details to the rate.
Add child category to rate	addChildCategory	Adds child category details to the rate.
Add category inputs to category	addCategoryInputs	Adds category inputs to the category.

RATEMANAGER INTERFACE

Constructor Detail

EsiRateManager

```
public EsiRateManager(int aSubscriber)
               throws java.lang.Exception
```

Constructor for EsiRateManager using LOCALHOST as the target server

Parameters:
`aSubscriber` - the licensed subscriber ID

Throws:
`java.lang.Exception`

EsiRateManager

```
public EsiRateManager(int aSubscriber,
                      java.lang.String aTargetServer)
               throws java.lang.Exception
```

Constructor for EsiRateManager

Parameters:
`aSubscriber` - the licensed subscriber ID
`aTargetServer` - target server name or IP address

Throws:
`java.lang.Exception`

Environment Items – getAvailableEnvironments, All Non-Secured

- **Business Functionality**
  - Gets all the available non-secured environments.
  - This ESI interfaces provides the ability for an external system to programmatically retrieve meta-data about all the available non-secured Environments in order for the system to programmatically invoke the other ESI deployment related API's.
- **Inputs**
  - No inputs
- **Outputs**
  - List of all non-secured EsiEnvironment objects.

Environment Items – getAvailableEnvironments, Secured

- **Business Functionality**
  - Gets all the available environments for a particular environment type.
  - This ESI interfaces provides the ability for an external system to programmatically retrieve meta-data about the available Environments in order for the system to programmatically invoke the other ESI deployment related API's and pass in the correct environment data object.
- **Inputs**
  - aType – the environment type
  - isIncludeSecure – Include Secured or Not
- **Outputs**
  - List of ESIEnvironment objects.

General – createLauchDefinition (Selected Version, Network Login)

- **Business Functionality**
  - Method creates a URL of an existing program version. This method uses the Network login for authentication. Requires custom authentication setup in RateManager Security. See Authentication.
- **Inputs**
  - aLine – Line ID
  - aProgram – Program ID
  - aProgramVersion – Program Version
- **Outputs**
  - Returns a URL string to launch a Program.

General – createLauchDefinition (First Version, Network Login)

- **Business Functionality**
  - Method creates a URL of the first program version. This method uses the Network login for authentication. Requires custom authentication setup in RateManager Security. See Authentication.
- **Inputs**
  - aLine – Line ID
  - aProgram – Program ID
- **Outputs**
  - Returns a URL string to launch a Program.

General – createLauchDefinition (Selected Version, RateManager Login)

- **Business Functionality**
  - Method creates a URL of an existing program version. This method uses the standard RateManager login for authentication.
- **Inputs**
  - aUser – non-network user ID
  - aLine – Line ID

  - aProgram – Program ID
  - aProgramVersion – Program Version
- **Outputs**
  - Returns a URL string to launch a Program.

General – createLauchDefinition (First Version, RateManager Login)

- **Business Functionality**
  - Method creates a URL of the first program version. This method uses the standard RateManager login for authentication.
- **Inputs**
  - aUser – non-network user ID
  - aLine – Line ID
  - aProgram – Program ID
- **Outputs**
  - Returns a URL string to launch a Program.

Global Items – createInput

- **Business Functionality**
  - Creates a new global level input in a particular Line.
- **Inputs**
  - EsiInput
- **Outputs**
  - Returns the ID of the newly created Input if succeeded or exception if failed.

Global Items – updateInput

- **Business Functionality**
  - Updates a global level input in a particular Line.
- **Inputs**
  - EsiInput
- **Outputs**
  - True if successful or false if failed.

Global Items – deleteInput

- **Business Functionality**
  - Deletes a global level input in a particular Line.
- **Inputs**
  - EsiInput
- **Outputs**
  - True if successful or false if failed.

Global Items – getInputsArray

- **Business Functionality**
  - Lists all the inputs of a particular Line.
- **Inputs**
  - EsiInput
- **Outputs**
  - Returns an array of EsiInputs that includes all the inputs of that Line.

Global Items – createCategory

- **Business Functionality**
  - Creates a category in a particular Line.
- **Inputs**
  - EsiCreateCategory
- **Outputs**
  - Returns the ID of the newly created Category if succeeded or exception if failed.

Global Items – createCategory with Inputs

- **Business Functionality**
  - Creates a category with inputs in a particular Line.
- **Inputs**
  - EsiCategory
  - List of EsiInput objects
- **Outputs**
  - No outputs.

Global Items – getCategoryXML

- **Business Functionality**
  - Gets an XML string of all the categories in a particular Line.
- **Inputs**
  - getCategoryXML
- **Outputs**
  - Return an XML string.

Global Items – updateCategory

- **Business Functionality**
  - Updates a category in a particular Line.
- **Inputs**
  - EsiCategory
- **Outputs**
  - True if successful exception if failed.

Global Items – deleteCategory

- **Business Functionality**
  - Deletes a category in a particular Line.
- **Inputs**
  - EsiCategory
- **Outputs**
  - True if successful exception if failed.

Program Items – createLine

- **Business Functionality**
  - Creates a new line of business. The line of business, if activated, is displayed on the RateManager navigation bar immediately. In order for any users to use the new line, groups must have access privileges granted.
- **Inputs**
  - EsiLine
- **Outputs**
  - Returns the ID of the newly created Line if succeeded or -1 if failed.

Program Items – updateLine

- **Business Functionality**
  - Updates a line of business.
- **Inputs**
  - EsiLine
- **Outputs**
  - True if successful or exception if failed.

Program Items – deleteLine

- **Business Functionality**
  - Deletes a line of business.
- **Inputs**
  - EsiLine
- **Outputs**
  - True if successful or exception if failed.

Program Items – createFolder

- **Business Functionality**
  - Creates a new folder in a particular Line.
- **Inputs**
  - EsiFolder
- **Outputs**
  - Returns an EsiFolder object.

Program Items – updateFolder

- **Business Functionality**
  - Updates the name or parent of an existing program folder. When you change the parent of any folder, the folders underneath are changed as well.
- **Inputs**
  - EsiFolder
- **Outputs**
  - Returns the updated Folder ID if succeeded or exception if failed.

Program Items – deleteFolder

- **Business Functionality**
  - Deletes an existing program folder. Make absolutely sure that the programs for the folder are no longer needed. Deleting a folder deletes all programs under that folder only. Subfolders and any programs in the subfolders are not deleted but will now be unlinked.
- **Inputs**
  - EsiFolder
- **Outputs**
  - Returns True if succeeded or exception if failed.

Program Items – getFoldersXML

- **Business Functionality**
  - Gets an XML string of available Folders of a particular Line.
- **Inputs**
  - getFoldersXML
- **Outputs**
  - Returns an XML string of available Folders of a specific Line.

Program Items – createProgram

- **Business Functionality**
  - Creates a new Insbridge Program in the folder of your choosing.
- **Inputs**
  - EsiProgram
- **Outputs**
  - EsiProgram object.

Program Items – updateProgram

- **Business Functionality**
  - Updates a Program in the folder of your choosing.
- **Inputs**
  - EsiProgram

- **Outputs**
  - True if successful or exception if failed.

Program Items – deleteProgram

- **Business Functionality**
  - Deletes an existing Insbridge Program. When a program is deleted, all data (variables, algorithms, sequencing, result groups and, if an auto program, driver assignment scenarios) and versions are deleted. Make sure this is the action you want to take. Locked programs cannot be deleted and returns a fail.
- **Inputs**
  - EsiProgram
- **Outputs**
  - True if successful or exception if failed.

Program Items – createProgramVersion

- **Business Functionality**
  - Creates a new version of an existing program.
- **Inputs**
  - EsiProgramVersion
- **Outputs**
  - Returns an EsiProgramVersion object.

Program Items – copyProgram

- **Business Functionality**
  - Copies an existing Insbridge program version to a new specified program version with the option to copy the definition only or copy the definition and all data.
    - **Copy Definitions Only:** This option copies all elements of the program with the exception of the data contained in mapped variables.
    - **Copy Definitions & All Data:** This option copies all elements of the program, including the data contained in mapped variables.
- **Inputs**
  - New Program Version
  - Source Program Version
  - Copy Option
- **Outputs**
  - Returns an EsiProgram object.

Program Items – updateProgramVersionLock

- **Business Functionality**
  - Locks or unlocks a program version.
- **Inputs**
  - EsiProgramVersion

  - aLockOption
- **Outputs**
  - True if successful or exception if failed.

Releases Items – createRelease

- **Business Functionality**
  - Creates a new Release.
- **Inputs**
  - aReleaseName – the release name
  - aReleaseDescription – the release description
  - aReleaseNotes – The release notes
  - aAdminEmail – the release admin email
  - aTargetEnvironment – the environment
- **Outputs**
  - EsiRelease object.

Releases Items – deleteRelease

- **Business Functionality**
  - Deletes an existing release.
- **Inputs**
  - aTargetRelease – the release being deleted
- **Outputs**
  - True if successful or exception if failed.

Releases Items – addReleaseProgram

- **Business Functionality**
  - Add a program version to a Release.
- **Inputs**
  - aTargetRelease – the release the program version is being added to
  - aTargetProgramVersion – the program version that is being added
- **Outputs**
  - True if successful or exception if failed.

Releases Items – removeReleaseProgram

- **Business Functionality**
  - Removes a program version from a Release.
- **Inputs**
  - aTargetRelease – the release the program version is being removed from
  - aTargetProgramVersion – the program version that is being removed
- **Outputs**
  - True if successful or exception if failed.

Releases Items – getReleasePrograms

- **Business Functionality**
    - Lists the programs in a Release.
- **Inputs**
    - aTargetRelease – the EsiRelease object being retrieved
- **Outputs**
    - The EsiRelease object including the programs contained.

Security Items – login

- **Business Functionality**
    - Logs a User in to the RateManager System.
- **Inputs**
    - none
- **Outputs**
    - none

Security Items – networkLogin

- **Business Functionality**
    - Logs a User in to the RateManager System using a Windows login.
- **Inputs**
    - none
- **Outputs**
    - none

Security Items – logout

- **Business Functionality**
    - Logs a User out of the RateManager System.
- **Inputs**
    - none
- **Outputs**
    - none

Security Items – createUser

- **Business Functionality**
    - Creates a new User.
- **Inputs**
    - EsiUser
- **Outputs**
    - Returns new user ID if successful or exception if failed.

Security Items – updateUser

- **<u>Business Functionality</u>**
  - Updates an existing User. Password Reset must be done inside RM.
- **<u>Inputs</u>**
  - EsiUse
- **<u>Outputs</u>**
  - Returns True if succeeded or False if failed.

Security Items – getUsers

- **<u>Business Functionality</u>**
  - Gets a list of all the available users for the subscriber.
- **<u>Inputs</u>**
  - No inputs
- **<u>Outputs</u>**
  - Returns a list of EsiUsers.

Security Items – getUserGroups

- **<u>Business Functionality</u>**
  - Gets all the available user groups.
- **<u>Inputs</u>**
  - No inputs
- **<u>Outputs</u>**
  - Returns a list of EsiUserGroup.

SRP Items – createSRP

- **<u>Business Functionality</u>**
  - Creates a SoftRater Package (SRP) in a particular environment.
  - ESI created SRP has the identifier “API” in the filename and shows an icon on IBFA SRP list.
- **<u>Inputs</u>**
  - aTargetProgram – the program the SRP to be created on
  - aProgram Version – the program version the SRP to be created on
  - aTargetEnvironmnet – the environment the SRP to be created on
- **<u>Outputs</u>**
  - Returns the newly created EsiSRP object.

Chapter 5

FRAMEWORK ADMINISTRATOR INTERFACE

Constructor Detail

EsiFrameworkAdministrator

```
public EsiFrameworkAdministrator(java.lang.String aApplicationTokenKey,
                                 int aSubscriber)
                          throws java.lang.Exception
```

Constructor for FrameworkAdministrator using LOCALHOST as the target server

Parameters:
`aApplicationTokenKey` - the Secure token to validate ESI session
`aSubscriber` - the Licenced Subscriber ID

Throws:
`java.lang.Exception`

EsiFrameworkAdministrator

```
public EsiFrameworkAdministrator(java.lang.String aApplicationTokenKey,
                                 int aSubscriber,
                                 java.lang.String aTargetServer)
                          throws java.lang.Exception
```

FrameworkAdministrator Constructor

Parameters:
`aApplicationTokenKey` - the Secure token to validate ESI session
`aSubscriber` - the Licenced Subscriber ID
`aTargetServer` - the Target server name or IP address

Throws:
`java.lang.Exception`

SRP Items – copySRP

- **Business Functionality**
  - Copies an SRP from the current environment to the selected environment. To determine the current environment, run the GetSRPList. This returns a list of Esi_SRP's. Every Esi_SRP has the environment and other information. CopySRP gets the current environment from the Esi_SRP object that you pass in.
- **Inputs**

  - aTargetEnvironment – The environment the SRP to be copied to.
  - aTargetSRP – The SRP to be copied.
- **Outputs**
  - True if succeeded or exception if failed.

SRP Items – deleteSRP

- **Business Functionality**
  - Deletes an SRP from disk. To determine (or control) the environment that the SRP is deleted from, run the GetSRPList to get the Esi_SRP's. Every Esi_SRP has the environment and other information. This tells you which environment the SRP is currently in.
  - Once deleted, this package cannot be reloaded.
- **Inputs**
  - aTargetSRP – The SRP to be deleted
- **Outputs**
  - True if succeeded or exception if failed

SRP Items – exportSRP

- **Business Functionality**
  - Exports an SRP from an IBFA environment and converts it to bytes.
  - You can use the bytes directly or save them to a physical file by using SaveFile() method.
  - To use the SRP bytes, use a straight binary read into a byte array.

    ```
    FileStream fs = File.OpenRead(@"C:\TEMP\abc.srp");
    byte[] srpBytes = new byte[fs.Length];
    fs.Read(srpBytes, 0, srpBytes.Length);
    ```
- **Inputs**
  - aTargetSRP – The SRP to be exported
- **Outputs**
  - The bytes of the SRP

SRP Items – getSRPList

- **Business Functionality**
  - Gets a list of all SRPs in a particular environment.
- **Inputs**
  - aTargetEnvironment – The environment from where the SRPs are to be retrieved.
  - aSRPType – theSRP type, 0=all SRPs; 1=ESI created SRPs
- **Outputs**
  - A list of EsiSRP objects.

SRP Items – getBytesFromFile

- **Business Functionality**
  - Gets the bytes from the file you pass in.

- **Inputs**
  - aFile – the file to get bytes from.
- **Outputs**
  - The bytes of the file.

SRP Items – importSRP

- **Business Functionality**
  - Imports SRP bytes into an IBFA environment.
- **Inputs**
  - aTargetEnviroment – The environment the SRP is to be imported into
  - aSRPBytes – The SRP bytes to be imported
  - aSRPFileName – The SRP file name to be saved as
- **Outputs**
  - True if succeeded or exception if failed

SRP Items – loadSRP

- **Business Functionality**
  - Loads an SRP into an IBFA environment so that the rating algorithms can be executed against it.
  - Loading a package takes the logic stored in the package and places it in the database so it can be executed.
  - Packages should be able to be loaded within an ESI Environment.
- **Inputs**
  - aTargetEnvironment – The environment the SRP is to be loaded in.
  - aTargetSRP – The SRP to be loaded.
- **Outputs**
  - True if succeeded or exception if failed.

SRP Items – moveSRP

- **Business Functionality**
  - Given an Esi_SRP object as well as an ESI Environment, the API moves the SRP from the current environment to the passed-in environment.
- **Inputs**
  - aTargetEnvironment – The environment the SRP is to be moved to.
  - aTargetSRP – The SRP to be moved.
- **Outputs**
  - True if succeeded or exception if failed.

SRP Items – saveFile

- **Business Functionality**
  - Saves the bytes you pass in to a physical file.
- **Inputs**

  - aData – The bytes to be saved
  - aPath – The location and name the bytes are to be saved
- **Outputs**
  - True if succeeded or exception if failed

SRP Items – unloadSRP

- **Business Functionality**
  - Unloads an SRP from a particular Environment. To determine (or control) the environment that the SRP is unloaded from, run the GetSRPList to get the Esi_SRP's. Every Esi_SRP has the environment and other information. This tells you which environment the SRP is currently in.
  - The package itself remains on disk until deleted. It is not necessary to unload a package before loading another package for the same program and version.
  - If a package is loaded for a program and version that already has a package loaded, the old package is unloaded before the new package is loaded.
- **Inputs**
  - aTargetSRP – The SRP to be unloaded.
- **Outputs**
  - True if succeed or exception if failed.

INPUTBATCH

Create New Rate

- **Business Functionality**
  - Creates new rate object to be added to an input batch. A rate object is the object representation of a rate request.
- **Inputs**
  - Rating details
  - Program details
  - Parent category
  - Child category
  - Category inputs
- **Outputs**
  - Rate Object

Add Batch to Runtime DB

- **Business Functionality**
  - Once all the rates have been added to the InputBatch, it can be added to the runtime database where the input batch will be picked up and rated against the SoftRater engine and results will be stored back in the database.
- **Inputs**
  - Rates
- **Outputs**
  - Batch ID

Execute Batch

- **Business Functionality**
  - When an InputBatch is added to the runtime database, it can be rated against the SoftRater engine.
- **Inputs**
  - Batch Id
- **Outputs**
  - Job

Retrieve Input Rates

- **Business Functionality**
  - A rate request after being added to the runtime database can be retrieved by the policy number.
  - Rate request can be retrieved either as an XML or as a rate object.
- **Inputs**
  - Batch Id
  - Policy Number

- **Outputs**
    - XML or Rate Object

Submit Insbridge XML from Memory

- **Business Functionality**
    - Insbridge XML from memory can be submitted as a batch to the runtime database.
- **Inputs**
    - Insbridge XML
- **Outputs**
    - Batch ID

Submit Insbridge XML Files from Disk

- **Business Functionality**
    - Insbridge XML files from disk can be submitted as a batch to the runtime database.
- **Inputs**
    - Insbridge XML files disk path
- **Outputs**
    - Batch ID

Submit Rate Object to Runtime DB

- **Business Functionality**
    - A rate object can be submitted as a batch to the runtime database.
- **Inputs**
    - Rate Object
- **Outputs**
    - Batch ID

Chapter 7

RESULT BATCH

Create New Result

- **Business Functionality**
  - Creates new result object to be added to an InputBatch. A rate object is the object representation of a rate request.
- **Inputs**
  - Rating details
  - Program details
  - Parent category
  - Child category
  - Category inputs
- **Outputs**
  - Result Object

Retrieve Results

- **Business Functionality**
  - A result can be retrieved by policy number.
  - A result can be retrieved either as an XML or as a result object.
- **Inputs**
  - Batch Id
  - Policy Number
- **Outputs**
  - XML or Result Object

Chapter 8

JOB

Retrieve Job Status

- **Business Functionality**
  - Once a Job is submitted for execution, the status of the Job can be retrieved here.
- **Inputs**
  - Batch Id
- **Outputs**
  - Status XML or Status Object

Stop the Job

- **Business Functionality**
  - Once a Job is submitted for execution, it can be stopped even if it is being executed.
- **Inputs**
  - Batch Id
- **Outputs**
  - status

Chapter 9

RATE OR RESULT

Add Rating Details

- **Business Functionality**
  - Adds the rating details to the rate object.
- **Inputs**
  - LOB
  - Policy No
  - Environment definition

Add Program Details

- **Business Functionality**
  - Adds the program details to the rate object
- **Inputs**
  - Parent ID
  - Program Id
  - Program Version

Add Parent Category

- **Business Functionality**
  - Adds the parent category in the rate object.
- **Inputs**
  - Category Id
  - Category Description

Add Child Category

- **Business Functionality**
- Adds child category to the current parent category.
- **Inputs**
  - Category ID
  - Category Description

Add Category Inputs

- **Business Functionality**
  - Adds the category inputs to the current category.
- **Inputs**
  - ID
  - Name
  - Value

Chapter 10

ESI USER SETUP AND AUTHENTICATION

There are two types of authentication in ESI RateManager Interface:

- **Regular login:** users send down a valid RateManager username and ESI logs the user in with that.
- **Network login:** ESI gets your windows credentials and logs you in with the same username and password as the user you login to windows

You must set up a network user account if you want to use windows integrated login in RateManager.

Adding a User

New users can be added from the User Management screen. All new users are assigned the default password, password. New users should log in as soon as possible and change their password.

NOTE: *Prior to creating a user, make sure you have your user groups created.*

To Add a New User

1. In RateManager, Tools→Security→**User Management**.
2. Click **NEW** to open the Add User tab.
3. Select the **Group**(s) the new user belongs to by selecting a group or groups from the Select Groups list and then clicking Select Arrow – Move Left. Upon saving, the user belongs to that group.
4. Select the **Company** the user belongs to from the drop down listing.

   Subscriber information may be needed by users when logging into the system. If more than one subscriber was created, users must be directed to the subscriber where their logins were created. Subscriber information is necessary for custom XML creation and testing and is also needed for Libraries.

   If you have not created subscribers in IBFA, you will not be able to assign users to a company.
5. Enter a **First** and **Last** name for the user in the appropriate text boxes.
6. Enter a **Username** for the new user. The username must be at least six, but fewer than twelve, alphanumeric characters and can include an underscore (_) or period (.).
7. Enter a **Department** for the user.
8. Select whether the new user is a **Network User** by selecting **True** or **False** from the drop down. If

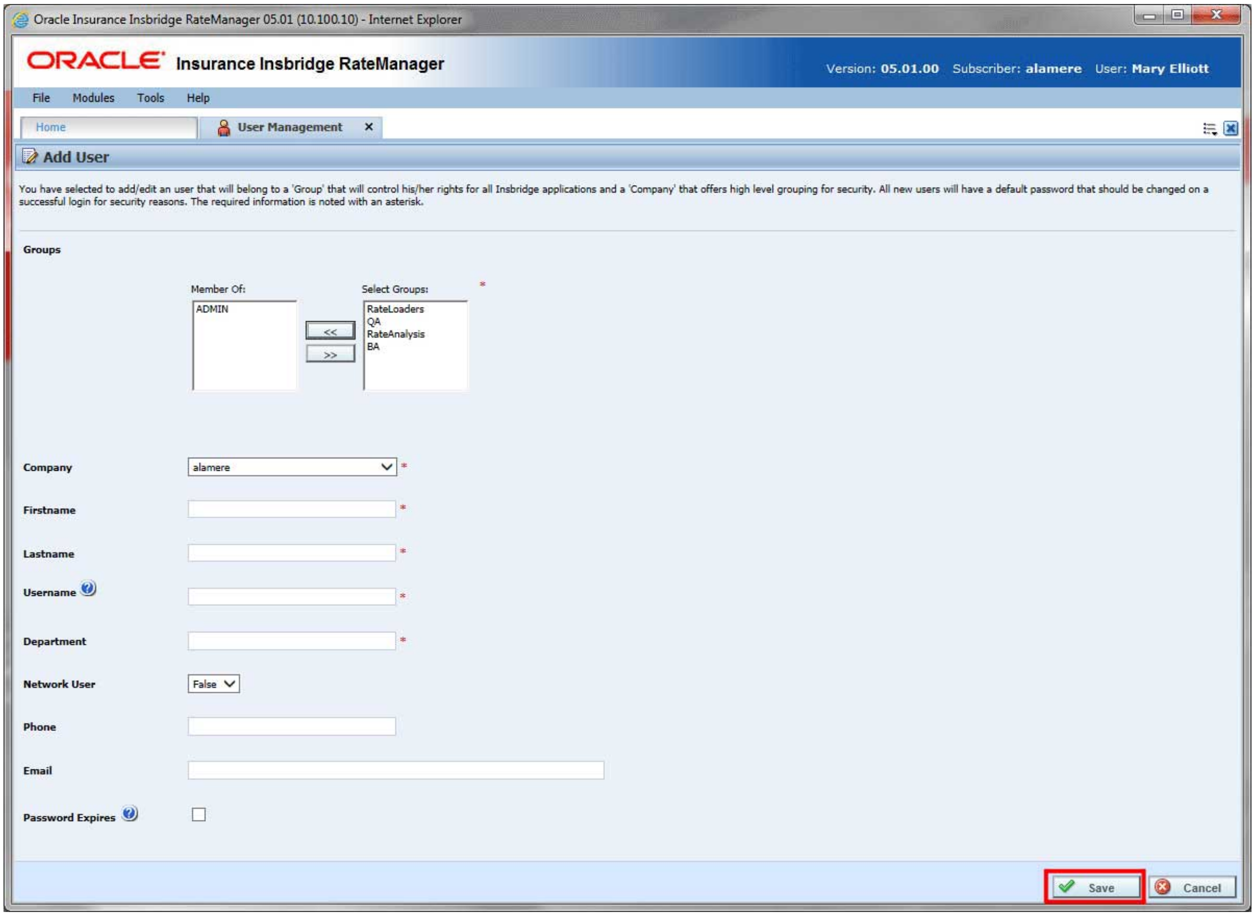
you select **True,** the user's credentials (username and password) are validated using Windows Authentication Tokens. The username must match the network user ID. If you select **False,** the user must enter a valid username and password via the RateManager login screen.

9. Optionally, enter a **Phone Number** and **Email Address** for the new user.

10. If you want the user's password to expire, click the box next to **Password Expires**. The Preferences screen contains the number of days required for a password change.

11. When you are finished, click **SAVE**. The user is added to the system and the user list is updated. The new user can now log in using the username you assigned them and the default password, **password**.

12. The user should change their password the first time they log in. See Changing Your Password for more information.


Oracle Insurance Insbridge RateManager 05.01 (10.100.10) - Internet Explorer
ORACLE Insurance Insbridge RateManager
Version: 05.01.00 Subscriber: alamere User: Mary Elliott
File
Modules
Tools
Help
Home
User Management
Add User
<<
>>
Save
Cancel

You have selected to add/edit an user that will belong to a 'Group' that will control his/her rights for all Insbridge applications and a 'Company' that offers high level grouping for security. All new users will have a default password that should be changed on a successful login for security reasons. The required information is noted with an asterisk.

Groups

Member Of: ADMIN

Select Groups: * RateLoaders, QA, RateAnalysis, BA

Company: alamere *

Firstname: *

Lastname: *

Username: *

Department: *

Network User: False

Phone:

Email:

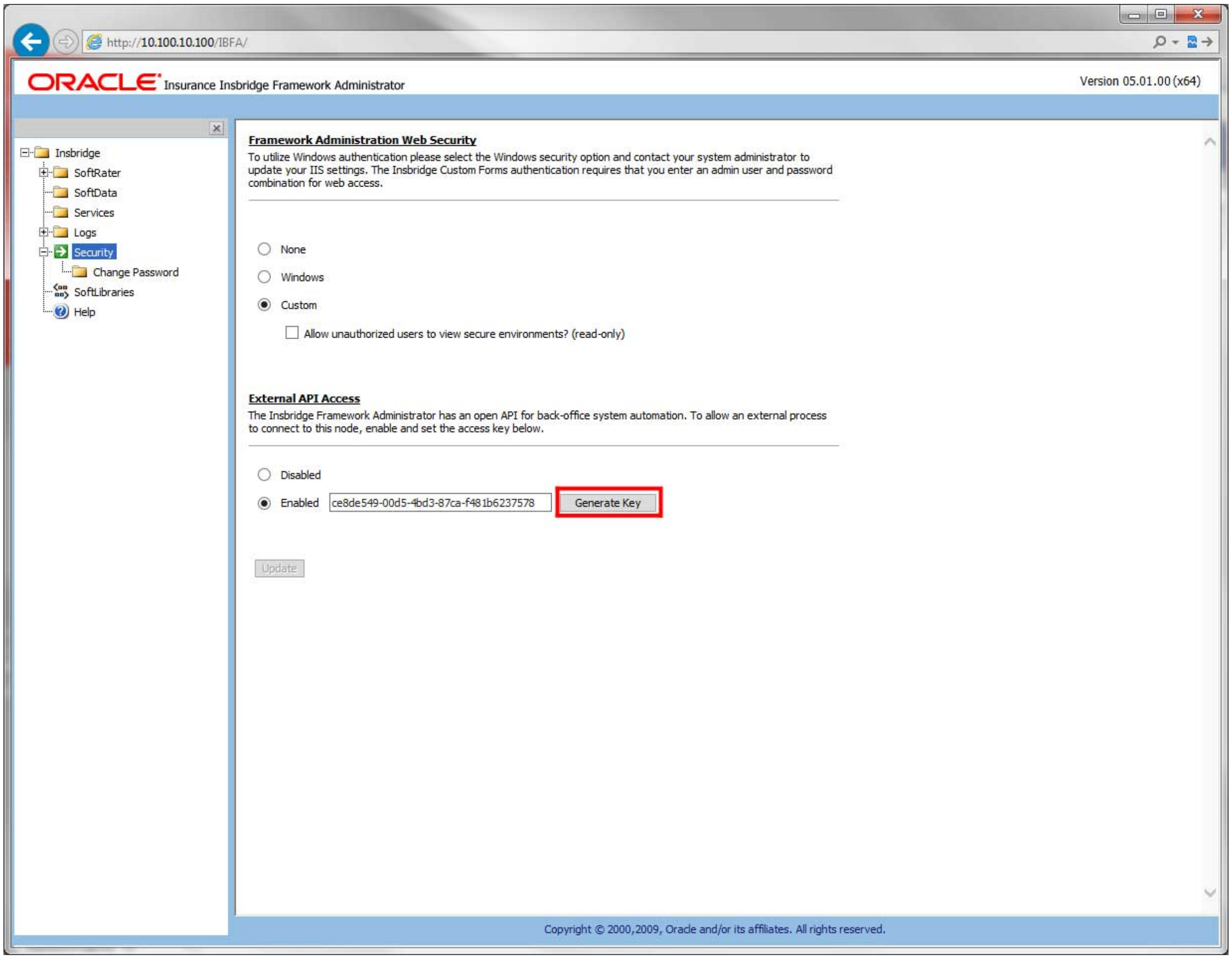
Password Expires: □

Figure 1 Adding a New User

ESI System Configuration

To use ESI's IBFA Framework functionality, you need to send down the API key when you do the ESI API call.

When IBFA is installed, a default API key is generated and saved in IBFA. You can find it in *IBFA→Insbridge→Security→External API Access*. You can disable ESI's IBFA functionality by setting it to "Disabled". For security purpose, you may want to change the key once in a while. To change the key, click on "Generate Key" and then click on "Update". Every key generated is unique.



Framework Administration Web Security

To utilize Windows authentication please select the Windows security option and contact your system administrator to update your IIS settings. The Insbridge Custom Forms authentication requires that you enter an admin user and password combination for web access.

○ None

○ Windows

● Custom

□ Allow unauthorized users to view secure environments? (read-only)

External API Access

The Insbridge Framework Administrator has an open API for back-office system automation. To allow an external process to connect to this node, enable and set the access key below.

○ Disabled

● Enabled ce8de549-00d5-4bd3-87ca-f481b6237578

Copyright © 2000,2009, Oracle and/or its affiliates. All rights reserved.

Figure 2 Allowing External API Access

ESI – IBSS System Configuration

List of Files

The following are the list of files needed to Test ESI - IBSS

File	Usage
EsiSwingTester.cmd	Command file that will start the swings tester Application.
EsiSwingTester.jar	EsiSwingTester– Swings App
InsbridgeESIServices.jar	Esi Services are packaged to this jar

Prerequisite:

- Set up JAVA_HOME correctly to the location where java is installed in the system.
- InsbridgeESIServices.jar must be in the same folder as the esiTester.cmd. If not, then it must be configured in the classpath variable of cmd file.

EsiSwingTester.cmd

This cmd file starts the esi-tester swings app.

EsiSwingTester.jar

- The tester swing app can help in testing the ESI.
- Both IBSS and IBFA testing are integrated. This document covers only the IBSS Testing. Please see ESI for Windows for more information.

InsbridgeESIServices.jar

- The ESI Logic and the services part are packaged in this .jar file.
- There is a config.properites file in this .jar that used for setting up the configuration for ESI.

Configuration Changes

The config.properties requires editing prior to usage.

Config.properties

1. Open the InsbridgeESIServices.Jar file using an unzip utility and open the config.properties file for editing.

2. Change the following to match your server and EJB details where IBSS is deployed.

   – ESI_IN_SERVER=false

   – hostName=localhost:7101

   – contextRoot=ins.ru.sr.ui-context-root

If ESI is deployed to an Application Server, then the following details need to be configured.

Weblogic Configuration

– ESI_EJB_JNDI=ESI#com.oracle.ins.ru.sr.bsn.ejb.esi.ESI

– INITIAL_CONTEXT_FACTORY=weblogic.jndi.WLInitialContextFactory

– PROVIDER_URL=t3://127.0.0.1:7103

WebSphere Configuration

– ESI_EJB_JNDI=com.oracle.ins.ru.sr.bsn.ejb.esi.ESI

– INITIAL_CONTEXT_FACTORY= com.ibm.websphere.naming.WsnInitialContextFactory

– PROVIDER_URL=iiop://127.0.0.1:7103

JBOSS Configuration

– ESI_EJB_JNDI= IBSS/ins.ru.sr.bsn-1.0.0/SoftRaterEJB!
  com.oracle.ins.ru.sr.bsn.ejb.esi.ESI

– INITIAL_CONTEXT_FACTORY= org.jnp.interfaces.NamingContextFactory

– PROVIDER_URL=jnp://127.0.0.1:7103

ESI IBSS TESTING

Submit Insbridge XML as XML

1. Open the Insbridge XML tab
2. Select Insbridge XML as XML Type
3. Select the test type as XML
4. Copy and paste the Insbridge XML to the Insbridge XML Text area
5. Enter the subscriber ID
6. Enter the Environment ID
7. Click on submit to Runtime DB
8. Result will be your batch id or Error message in case of errors

Submit Insbridge XML as Objects

1. Open the Insbridge XML tab
2. Select insbridge XML as XML Type
3. Select the test type as Object
4. Copy and paste the Insbridge XML to the Insbrdige XML Text area
5. Enter the subscriber ID
6. Enter the Environment ID
7. Click on submit to Runtime DB
8. Result will be your batch id or Error message in case of errors

Submit Insbridge XML Files to Queue

1. Open the Insbridge XML tab
2. Select File as XML Type
3. Select the test type as XML
4. Enter the path to the location of the xml files
5. Enter a name for the batch
6. Enter the subscriber ID
7. Enter the Environment ID

8. Click on submit to Runtime DB
9. Check Broker messages with status as pending to see if the files are added to the queue.
10. Result will be your batch id or Error message in case of errors

Submit Insbridge XML Files as Objects To Queue

1. Open the Insbridge XML tab
2. Select XML Type as File
3. Select the test type as Object
4. Enter the path to the location of the xml files
5. Enter a name for the batch
6. Enter the subscriber ID
7. Enter the Environment ID
8. Click on submit to Runtime DB
9. Check Broker messages with status as pending to see if the files are added to the queue.
10. Result will be your batch id or Error message in case of errors

Database Functions

ESI currently supports 3 database related functions as listed below:

Get Input XML by Policy Number

1. Open the Insbridge XML tab
2. Select XML Type as DB
3. Select Get Input XML as DB Method
4. Select the test type as XML
5. Enter the batch id and Policy Number
6. Enter the subscriber ID
7. Enter the Environment ID
8. Click on submit to Runtime DB
9. Result will be your input xml or Error message in case of errors

Get Input XML by policy Number as Object

1. Open the Insbridge XML tab
2. Select XML Type as DB
3. Select Get Input XML as DB Method
4. Select the test type as Object
5. Enter the batch id and file Policy Number
6. Enter the subscriber ID
7. Enter the Environment ID
8. Click on submit to Runtime DB
9. Result will be your input xml or Error message in case of errors

Get Result XML by Policy Number

1. Open the Insbridge XML tab
2. Select DB as XML Type
3. Select Get Result XML as DB Method
4. Select the test type as XML
5. Enter the batch id and Policy Number
6. Enter the subscriber ID
7. Enter the Environment ID
8. Click on submit to Runtime DB
9. Result will be your Result xml or Error message in case of errors

Get Result XML by Policy Number as Object

1. Open the Insbridge XML tab
2. Select DB as XML Type
3. Select Get Result XML as DB Method
4. Select the test type as Object
5. Enter the batch id and Policy Number
6. Enter the subscriber ID
7. Enter the Environment ID

8. Click on submit to Runtime DB
9. Result will be your Result xml or Error message in case of errors

Copy Batch

1. Open the Insbridge XML tab
2. Select DB as XML Type
3. Select Copy Batch as DB Method
4. Select the test type as XML
5. Enter the batch id
6. Enter the subscriber ID
7. Enter the Environment ID
8. Click on submit to Runtime DB
9. Result will be your New Batch ID or Error message in case of errors

BATCH – STARTING AND MONITORING

Start Batch

1. Enter Batch ID, Subscriber ID, Environment ID, File Ref id, Notification details and read write option.
2. Click on start Batch
3. Result will be your request id or error in case of errors

Stop Batch

1. Enter request Id
2. Click on Stop Batch
3. Result will be the status of the stop batch request

Get Batch Status

1. Enter request Id

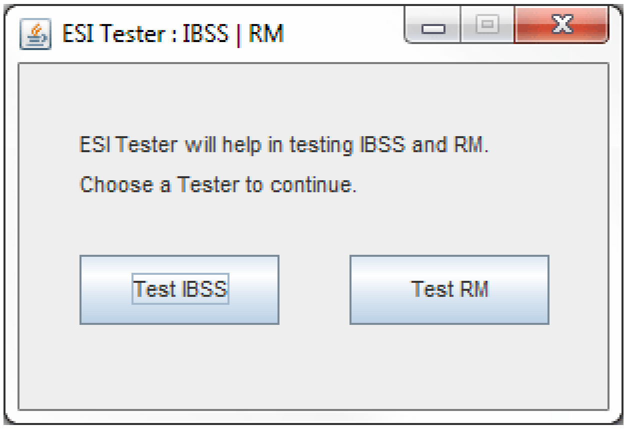
2. Click on Get Batch Status

3. Result will be the status of the batch request

Chapter 12

ESI TESTER

To use the ESI Tester, launch the .cmd file. The ESI tester runs on a Windows machine. The Java version of ESI allows for IBSS testing as well as RateManager. Select the option you want to test.



ESI Tester will help in testing IBSS and RM.

Choose a Tester to continue.

Figure 3 Selecting Instance to Test

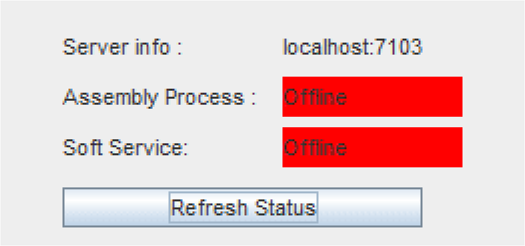
IBSS FEATURES/METHODS

The IBSS features.

- Insbridge XML
- XCategory
- Batch

Status is displayed in the upper corner. Offline indicates that webservices is not available. Check the configuration settings to make sure that correct configuration is being used. Refresh to update the status.

If the incorrect configuration for the application server is used, that status displays as offline.

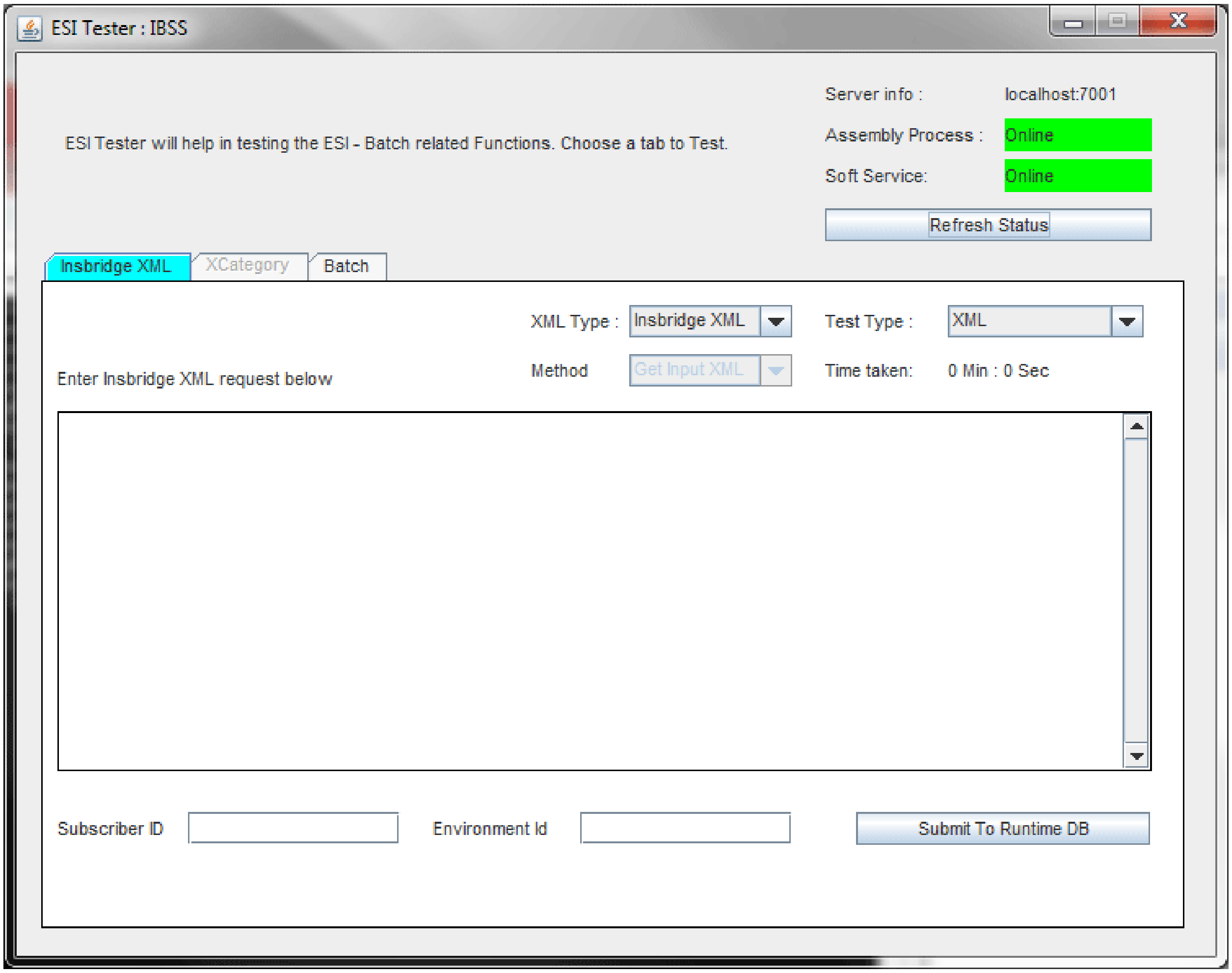


Server info : localhost:7103

Assembly Process : Offline

Soft Service: Offline

Figure 4 Connection Status



ESI Tester : IBSS

ESI Tester will help in testing the ESI - Batch related Functions. Choose a tab to Test.

Server info : localhost:7001

Assembly Process : Online

Soft Service: Online

XML Type : Insbridge XML

Test Type : XML

Method Get Input XML

Time taken: 0 Min : 0 Sec

Enter Insbridge XML request below

Subscriber ID

Environment Id

Figure 5 Testing IBSS

Testing IBSS allows for:

- Submit Insbridge XML as XML
- Submit Insbridge XML as Objects
- Submit Insbridge XML Files to Queue
- Submit Insbridge XML Files as Objects To Queue

Database Functions

ESI currently supports database related functions:

- Get Input XML by Policy Number
- Get Input XML by policy Number as Object
- Get Result XML by Policy Number
- Get Result XML by Policy Number as Object
- Copy Batch

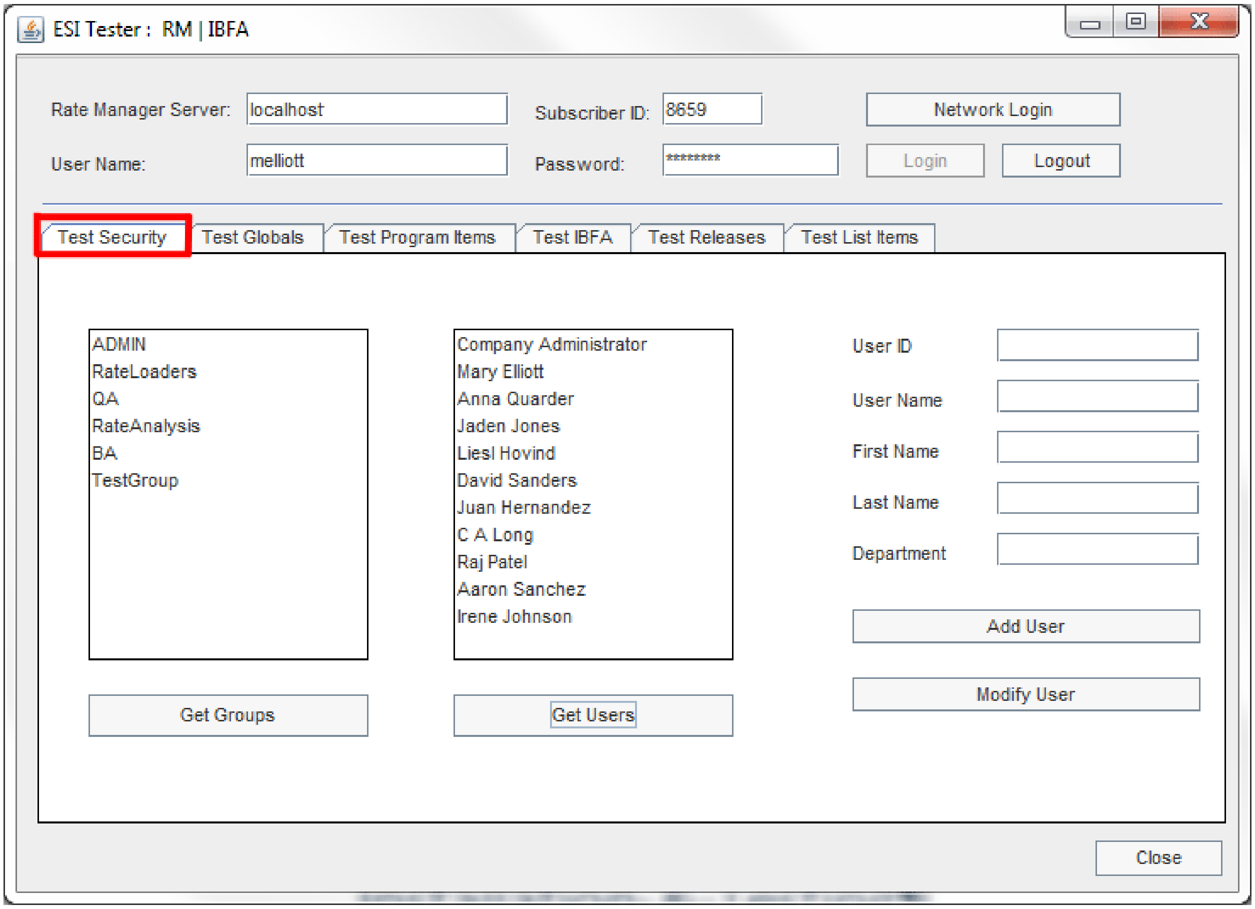
Select the types and paste in the XML request. Enter in Subscriber and environment Id. Click Submit to Runtime DB.

RM FEATURES/METHODS

- Test Security
- Test Globals
- Test Program Items
- Test IBFA
- Test Releases
- Test List Items

Test Security

Allows for the adding and editing of User Groups and Users.



ESI Tester : RM | IBFA

Rate Manager Server: localhost

Subscriber ID: 8659

User Name: melliott

Password: ********

ADMIN
RateLoaders
QA
RateAnalysis
BA
TestGroup

Company Administrator
Mary Elliott
Anna Quarder
Jaden Jones
Liesl Hovind
David Sanders
Juan Hernandez
C A Long
Raj Patel
Aaron Sanchez
Irene Johnson

User ID

User Name

First Name

Last Name

Department

Figure 6 Test Security

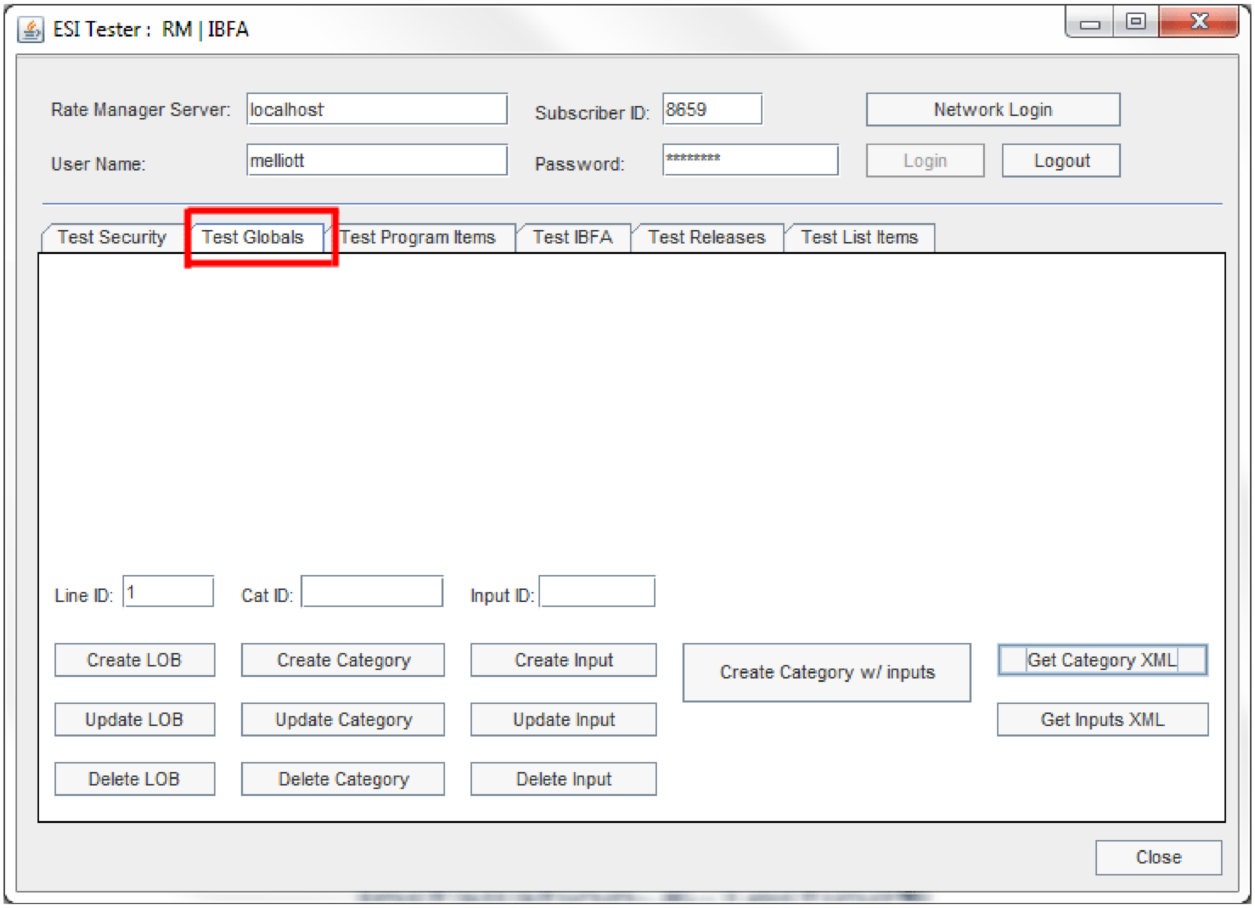
- Get Groups returns the current groups in the system
- Get Users returns a list of the current users in the system.
- Add User and Modify User requires**:**
    - User Name

- First Name
- Last Name

Other fields are optional

Test Globals

Allows for the adding, editing and deleting of LOBs, Categories, and Inputs.



ESI Tester : RM | IBFA

Rate Manager Server: localhost

Subscriber ID: 8659

User Name: melliott

Password: ********

Line ID: 1

Cat ID:

Input ID:

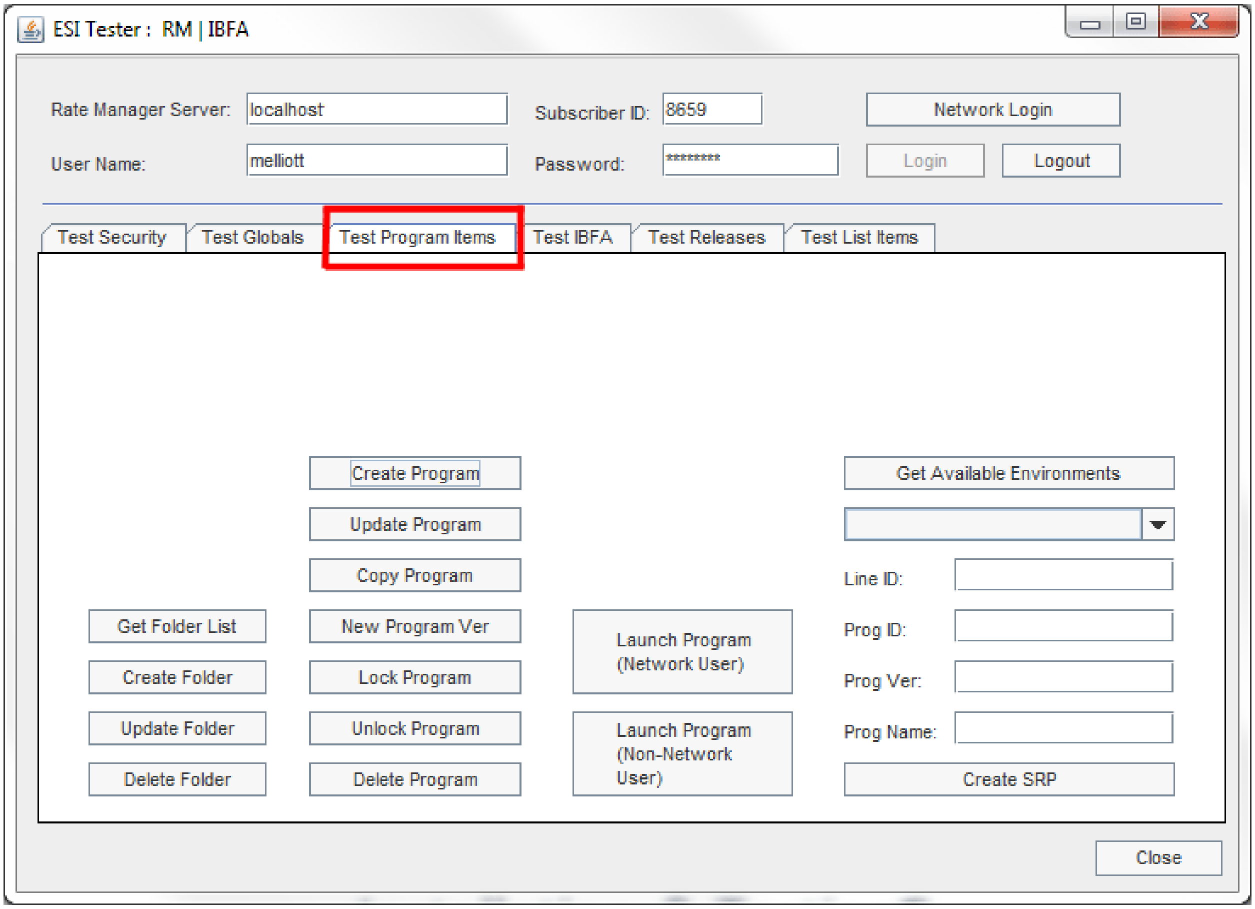
Figure 7 Test Globals

LOB requests require a Line ID.

- Get Category XML returns the details of the categories in the selected line.
- Get Inputs XML returns the details of the inputs in the selected line.

Test Program Items

Allows for viewing, adding, editing, copying and deleting of program items.



ESI Tester : RM | IBFA

Rate Manager Server: localhost

Subscriber ID: 8659

User Name: melliott

Password: ********

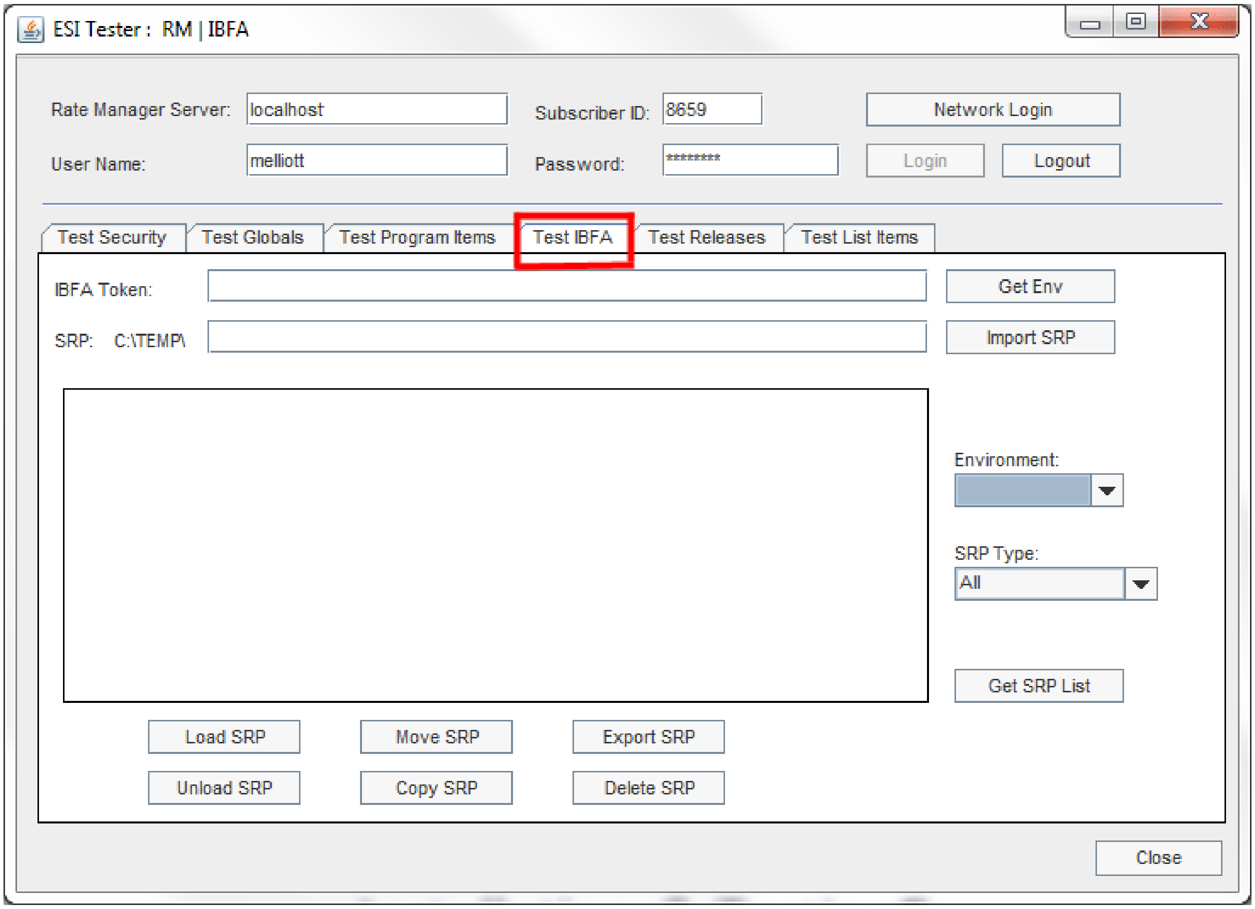
Figure 8 Test Program Items

Get Available Environments returns the environments available. The list is populated drop down.

Get Folder List requires a Line ID and returns a list of folders in the line.

Test IBFA

Allows for SRPs to be managed.



ESI Tester : RM | IBFA

Rate Manager Server: localhost

Subscriber ID: 8659

User Name: melliott

Password: ********

SRP: C:\TEMP\

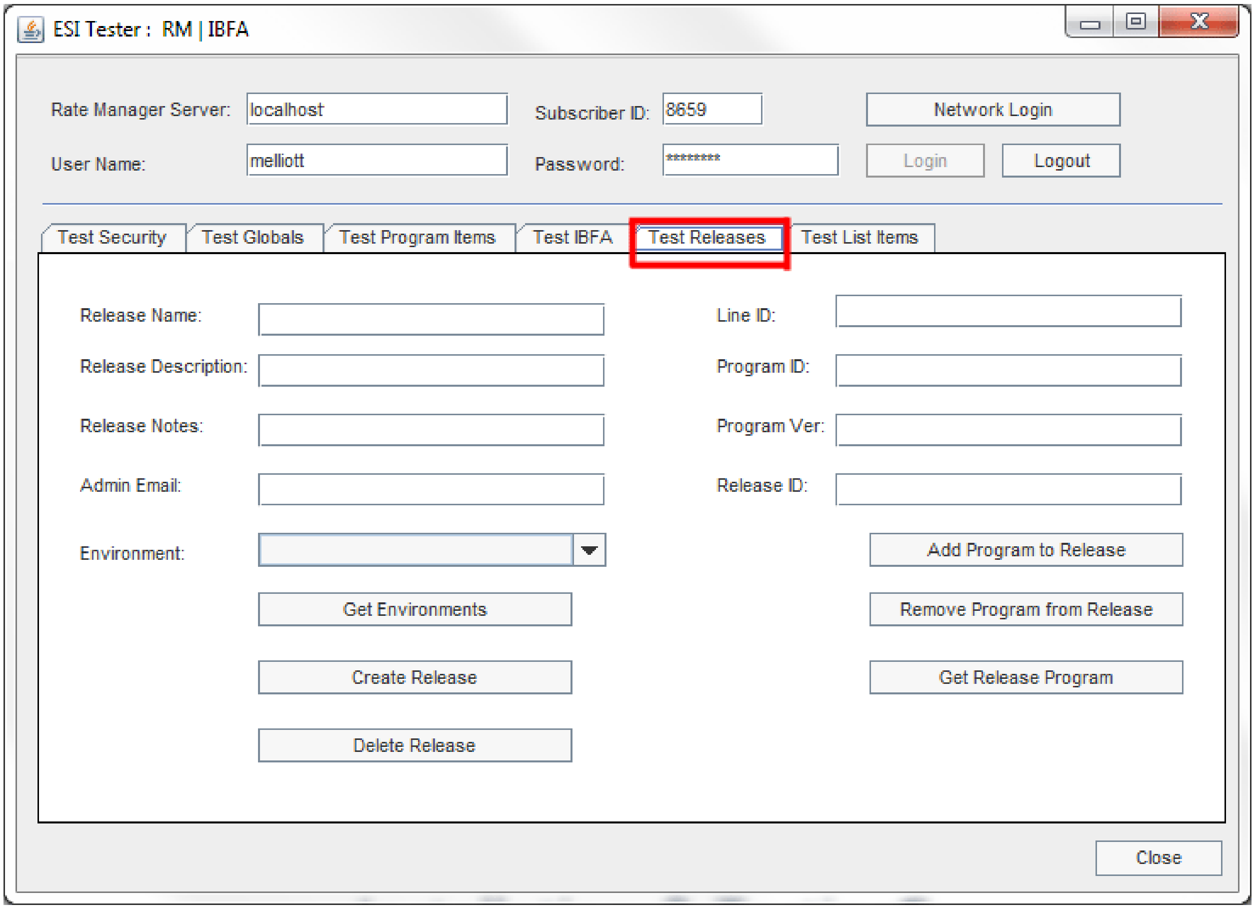
SRP Type: All

Figure 9 Test IBFA

Get Env returns the environments available. The list is populated Environment drop down.

Test Releases

Allows for the management of releases.



ESI Tester : RM | IBFA

Rate Manager Server: localhost

Subscriber ID: 8659

User Name: melliott

Password: ********

Figure 10 Test Releases

Get Environments returns the environments available. The list is populated Environment drop down.

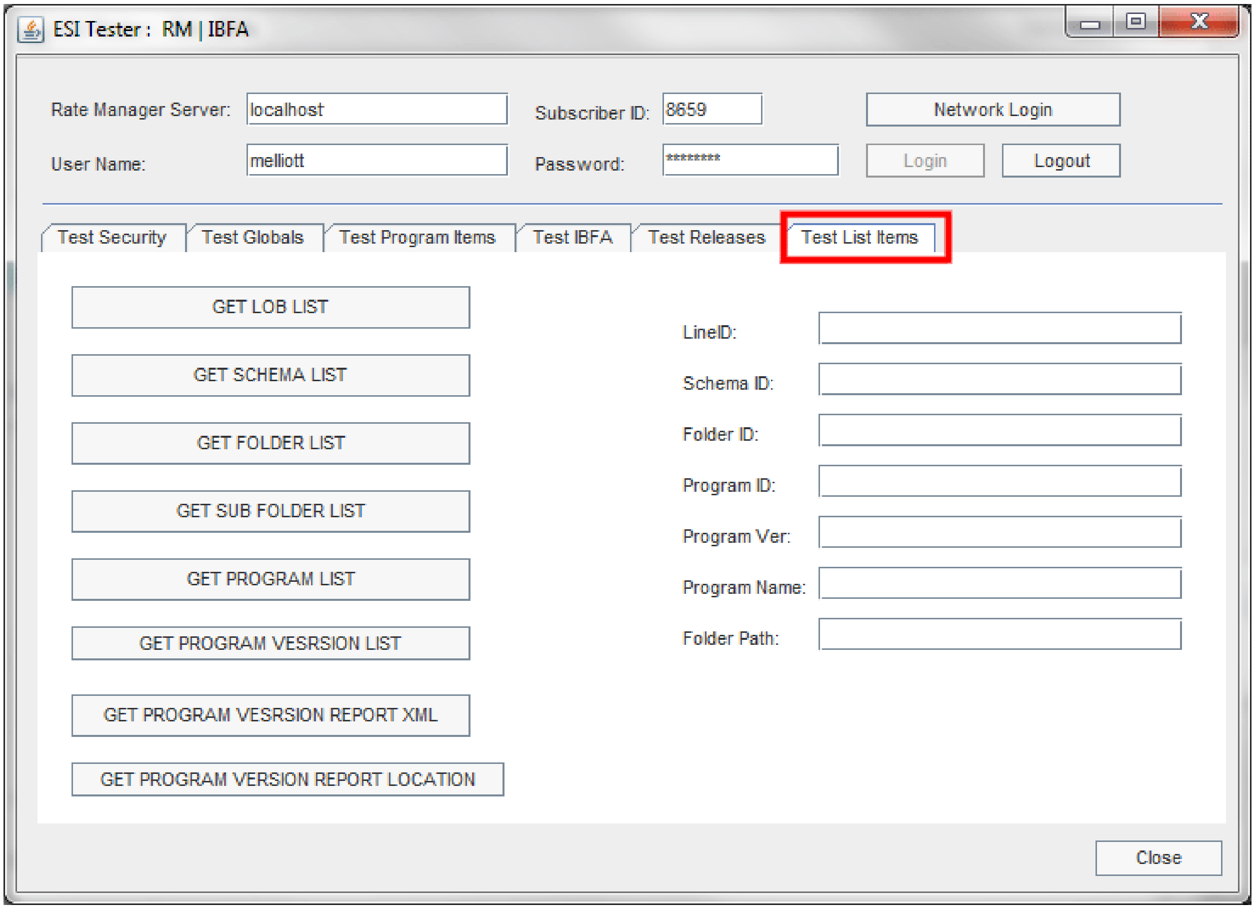
Test List Items

- Return a list of lines of business
- Return a list of schemas for a particular line of business
- Return a list of subline (schema) folders for a particular schema
- Return a list of folder for a particular subline folder
- Return a list of programs for a subline folder
- Return a list of program versions for a particular program
- Return the XML for the program version report for a particular program version

NOTE: *For the above all features it will return the list in the XML format.*

Test List Items allows you to build a treeview that follows the program explorer in RateManager and for the XML used for the program version report to be returned.

The columns on the right follow the treeview found on the Program Explorer in RateManager.



Rate Manager Server: localhost
Subscriber ID: 8659
User Name: melliott
Password: ********

Figure 11 Java ESI Tester for RM

To Use

Options are available after you enter in the Subscriber ID, username and password.

Enter in any required input and click the button of the items where you want returned. The output is returned in XML format.

GET LOB LIST

- **Input:** No input is required.
- **GET LOB LIST:** Gets an XML string of all the line of business including active and inactive lobs.
- **Output:** Returns an XML string of available LOB's. XML string has:
  - LOB_ ID
  - LOB_NAME
  - ACTIVE attributes

GET SCHEMA LIST

- **Inputs:** Line ID.
- **GET SCHEMA LIST:** Gets an XML string of available schemas of a particular LOB.
- **Outputs:** Returns an XML string that includes all the schemas of that LOB. XML string has:

- LOB_ID
- SCHEMA_ID
- SCHEMA_ NAME

GET FOLDER LIST

- **Inputs:** Line ID and Schema ID.
- **GET FOLDER LIST:** Gets an XML string of available folders of a particular Schema.
- **Outputs:** Returns the XML string that includes all the folders of that Schema. XML String has:

- LOB_ID
- SCHEMA_ID
- FOLDER_ID
- FOLDER_ NAME

GET SUB FOLDER LIST

- **Inputs:** Line ID, Schema ID and Folder ID.
- **GET SUB FOLDER LIST:** Gets an XML string of available sub folders of a particular subline folder/ folder.
- **Outputs:** Returns an xml string that includes all the sub folders of that Subline Folder/Folder. XML string has

- LOB_ID
- SCHEMA_ID
- FOLDER_ID
- FOLDER_ NAME
- FOLDER_PARENT_ID

GET PROGRAM LIST

- **Inputs:** Line ID, Schema ID and Folder ID.

- **GET PROGRAM LIST:** Gets an XML string of available Programs of a particular folder.

- **Outputs:** Returns an xml string that includes all programs of that Folder.
  XML string has:

  - LOB_ID
  - SCHEMA_ID
  - FOLDER_ID
  - PROGRAM_ ID
  - PROGRAM_ NAME

GET PROGRAM VERSIONS LIST

- **Inputs:** Line ID, Schema ID, Folder ID and Program ID.

- **GET PROGRAM VERSIONS:** Gets an XML string of available Program versions of a particular program.

- **Outputs:** Returns an xml string that includes all program versions of that Program.

XML string has:

- LOB_ID
- SCHEMA_ID
- FOLDER_ID
- PROGRAM_ ID
- VERSION

GET PROGRAM VERSION REPORT XML

- **Inputs:** Line ID, Schema ID, Folder ID, Program ID, Program Ver and Program Name.

- **GET PROGRAM VER REPORT XML:** Gets an XML string of program version report of a particular program version.

- **Outputs:** Returns an XML string of program version report.

GET PROGRAM VERSION REPORT LOCATION

This method returns the saved location of the program version report.

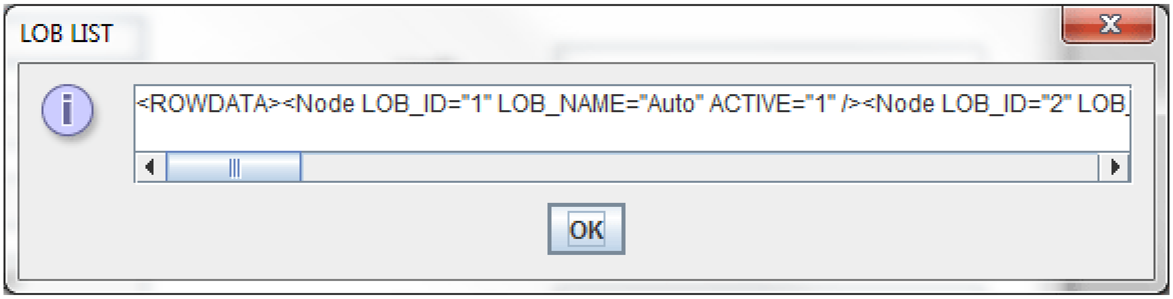
- **Inputs:** Line ID, Schema ID, Folder ID, Program ID, Program Ver , Program Name and Folder Path(optional).

- **Outputs:** Returns the saved location of the program version report.

NOTE: *The input Folder Path is optional. Folder Path is the Folder/Directory location of the report where it has to save. If we give the folder path it will generate the report and saves in the given location. If we don't give the Folder Path then by default system will create one location and it will save that location.*

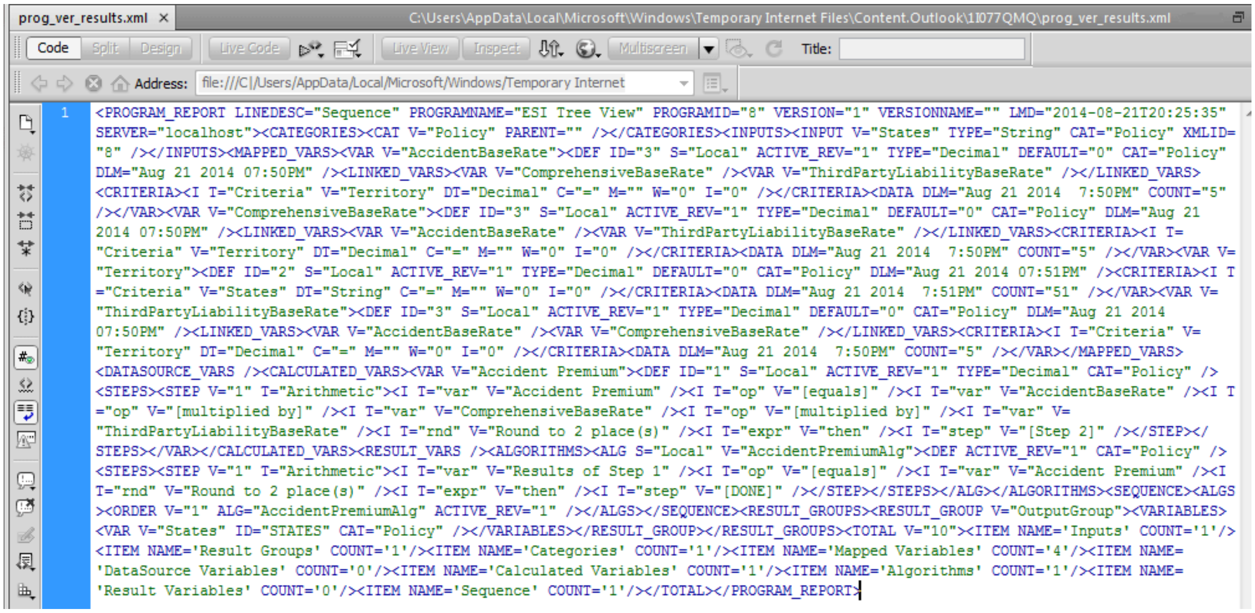
Output

Output is in XML format.



```
<ROWDATA><Node LOB_ID="1" LOB_NAME="Auto" ACTIVE="1" /><Node LOB_ID="2" LOB_
```

Figure 12 Tester Output



prog_ver_results.xml
C:\Users\AppData\Local\Microsoft\Windows\Temporary Internet Files\Content.Outlook\1I077QMQ\prog_ver_results.xml
Code
Split
Design
Live Code
Live View
Inspect
Multiscreen
Title:
Address:
file:///C|/Users/AppData/Local/Microsoft/Windows/Temporary Internet

```
<PROGRAM_REPORT LINEDESC="Sequence" PROGRAMNAME="ESI Tree View" PROGRAMID="8" VERSION="1" VERSIONNAME="" LMD="2014-08-21T20:25:35"
SERVER="localhost"><CATEGORIES><CAT V="Policy" PARENT="" /></CATEGORIES><INPUTS><INPUT V="States" TYPE="String" CAT="Policy" XMLID=
"8" /></INPUTS><MAPPED_VARS><VAR V="AccidentBaseRate"><DEF ID="3" S="Local" ACTIVE_REV="1" TYPE="Decimal" DEFAULT="0" CAT="Policy"
DLM="Aug 21 2014 07:50PM" /><LINKED_VARS><VAR V="ComprehensiveBaseRate" /><VAR V="ThirdPartyLiabilityBaseRate" /></LINKED_VARS>
<CRITERIA><I T="Criteria" V="Territory" DT="Decimal" C="=" M="" W="0" I="0" /></CRITERIA><DATA DLM="Aug 21 2014  7:50PM" COUNT="5"
/></VAR><VAR V="ComprehensiveBaseRate"><DEF ID="3" S="Local" ACTIVE_REV="1" TYPE="Decimal" DEFAULT="0" CAT="Policy" DLM="Aug 21
2014 07:50PM" /><LINKED_VARS><VAR V="AccidentBaseRate" /><VAR V="ThirdPartyLiabilityBaseRate" /></LINKED_VARS><CRITERIA><I T=
"Criteria" V="Territory" DT="Decimal" C="=" M="" W="0" I="0" /></CRITERIA><DATA DLM="Aug 21 2014  7:50PM" COUNT="5" /></VAR><VAR V=
"Territory"><DEF ID="2" S="Local" ACTIVE_REV="1" TYPE="Decimal" DEFAULT="0" CAT="Policy" DLM="Aug 21 2014 07:51PM" /><CRITERIA><I T
="Criteria" V="States" DT="String" C="=" M="" W="0" I="0" /></CRITERIA><DATA DLM="Aug 21 2014  7:51PM" COUNT="51" /></VAR><VAR V=
"ThirdPartyLiabilityBaseRate"><DEF ID="3" S="Local" ACTIVE_REV="1" TYPE="Decimal" DEFAULT="0" CAT="Policy" DLM="Aug 21 2014
07:50PM" /><LINKED_VARS><VAR V="AccidentBaseRate" /><VAR V="ComprehensiveBaseRate" /></LINKED_VARS><CRITERIA><I T="Criteria" V=
"Territory" DT="Decimal" C="=" M="" W="0" I="0" /></CRITERIA><DATA DLM="Aug 21 2014  7:50PM" COUNT="5" /></VAR></MAPPED_VARS>
<DATASOURCE_VARS /><CALCULATED_VARS><VAR V="Accident Premium"><DEF ID="1" S="Local" ACTIVE_REV="1" TYPE="Decimal" CAT="Policy" />
<STEPS><STEP V="1" T="Arithmetic"><I T="var" V="Accident Premium" /><I T="op" V="[equals]" /><I T="var" V="AccidentBaseRate" /><I T
="op" V="[multiplied by]" /><I T="var" V="ComprehensiveBaseRate" /><I T="op" V="[multiplied by]" /><I T="var" V=
"ThirdPartyLiabilityBaseRate" /><I T="rnd" V="Round to 2 place(s)" /><I T="expr" V="then" /><I T="step" V="[Step 2]" /></STEP></
STEPS></VAR></CALCULATED_VARS><RESULT_VARS /><ALGORITHMS><ALG S="Local" V="AccidentPremiumAlg"><DEF ACTIVE_REV="1" CAT="Policy" />
<STEPS><STEP V="1" T="Arithmetic"><I T="var" V="Results of Step 1" /><I T="op" V="[equals]" /><I T="var" V="Accident Premium" /><I
T="rnd" V="Round to 2 place(s)" /><I T="expr" V="then" /><I T="step" V="[DONE]" /></STEP></STEPS></ALG></ALGORITHMS><SEQUENCE><ALGS
><ORDER V="1" ALG="AccidentPremiumAlg" ACTIVE_REV="1" /></ALGS></SEQUENCE><RESULT_GROUPS><RESULT_GROUP V="OutputGroup"><VARIABLES>
<VAR V="States" ID="STATES" CAT="Policy" /></VARIABLES></RESULT_GROUP></RESULT_GROUPS><TOTAL V="10"><ITEM NAME='Inputs' COUNT='1'/>
<ITEM NAME='Result Groups' COUNT='1'/><ITEM NAME='Categories' COUNT='1'/><ITEM NAME='Mapped Variables' COUNT='4'/><ITEM NAME=
'DataSource Variables' COUNT='0'/><ITEM NAME='Calculated Variables' COUNT='1'/><ITEM NAME='Algorithms' COUNT='1'/><ITEM NAME=
'Result Variables' COUNT='0'/><ITEM NAME='Sequence' COUNT='1'/></TOTAL></PROGRAM_REPORT>
```

Figure 13 Program Result XML File

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