An Australian Government Initiative

Standard Business Reporting

# SBR REFERENCE CLIENT DOCUMENTATION

## JAVA VERSION

Software Release v2.1.2

Purpose: This document describes the SBR Reference Client's features and functionality.

For further information or questions, contact the SBR Service Desk at **SBRServiceDesk@sbr.gov.au** or call **1300 488 231**

## DOCUMENT APPROVALS

***This document was approved by:***

| Title | Date |
|---|---|
| SBR Chief Solution Architect | June 2015 |

## COPYRIGHT

# TABLE OF CONTENTS

# TABLE OF FIGURES

# 1. INTRODUCTION

## 1.1. PURPOSE

The purpose of this document is to describe the functionality of the Java version of the SBR Reference Client to assist Software Developers in understanding what the Reference Client contains and how it works.

The main focus is on identifying the key components of the solution and highlighting the essential areas where the SBR SDK components are used.

## 1.2. AUDIENCE

This document is intended for use by Software Developers who are integrating Java business applications with SBR Core Services.

It assumes knowledge of software development in Java, SOAP Web services including Web service security concepts etc.

Developers should already be familiar with SBR concepts including the SBR service WSDLs and have read the SBR Web Services Implementation Guide (WIG).

## 1.3. REFERENCE DOCUMENTS

| Ref | Link | Description |
|---|---|---|
| 1. | SBR Glossary<br><br>https://www.sbr.gov.au/content/public/help/glossary | Glossary of terms used in the SBR Program. |
| 2. | SBR Developers SDK Guide<br><br>On the SBR website<br><br>https://www.sbr.gov.au/content/myhome/softwaredevelopers/sdkguide | Contains links to a number of documents which provide essential information required to understand the SBR SDK. |
| 3. | Web Services Implementation Guide (WIG)<br><br>Available from the Common Components Download page on the SBR website under Web Services<br><br>https://www.sbr.gov.au/content/myhome/softwaredevelopers/downloads/commoncomponents | Essential reading to understand SBR Core Services. |
| 4. | SBR Core Services Requester User Guide – Common | Describes the purpose, functionality and context of the SBR Core Services Requester. |

| | | |
|---|---|---|
| 5. | User Guide – ABR.SecurityTokenManager<br><br>Included in the ABR.SecurityTokenManager package | User guide for the ABR Security Token manager (ABR SecurityTokenManager API)<br><br>Available in .NET and Java versions. |
| 6. | ABR Authentication Client SDK Developer Guide<br><br>Included in the ABR AUSkey package | User guide for the ABR AUSkey Manager (ABR.SecurityTokenManager API)<br><br>Available in .NET and Java versions. |
| 7. | Agency Message Implementation Guides (MIGs)<br><br>Available from the SBR Website Downloads by agency page<br><br>https://www.sbr.gov.au/content/myhome/softwaredevelopers/downloads/downloadsbyagency | Provide information specific to reporting requirements for agencies. |
| 8. | SBR Taxonomy<br><br>Available from the SBR Website common components download page<br><br>https://www.sbr.gov.au/content/myhome/softwaredevelopers/downloads/commoncomponents | Describes the SBR reportable items and how these relate to agency reporting obligations. |

# 2. THE SBR REFERENCE CLIENT

## 2.1. OVERVIEW

The SBR Reference Client is aimed at giving Software Developers an example of how to connect to the SBR Core services using the SDK components provided by SBR. It demonstrates how to interact with an AUSkey store, use a credential from the store to call the VANguard Security Token Service, submit requests and interpret the responses received.

## 2.2. FEATURES

This version of the Java SBR Reference Client demonstrates the following aspects

- ABR AUSkey Manager (ABR.SecurityTokenManager Keystore API) to access stored credentials
- ABR Security Token Manager (ABR.SecurityTokenManager Request API) to connect to the Secure Token Service
- SBR Core Services Requester component to connect to SBR Core services

It also demonstrates the Error management techniques detailed in section 4 of the WIG.

It has the following caveats

- it does not demonstrate creation or validation of XBRL instances, it assumes that a validated XBRL instance is already available to be included into the request body.

### 2.3. FORMS INCLUDED IN THE SBR REFERENCE CLIENT

The SBR Reference Client contains Form information for a small subset of SBR Forms which are available for Software Developer testing. The set of forms included with the SBR Reference Client is an example only and is not intended to be a complete list of available forms.

It is also important to note that not all Agencies support 24x7 availability in the testing environment.

As the Form metadata included with the SBR Reference Client represents a subset of the form versions supported at the time of release, and as the forms and their versions evolve over time, it is recommended that developers contact the SBR testing team to check whether specific forms are still current and hence available for testing.

The values defined in the forms metadata may change with Agency updates so the values provided in the SBR Reference Client should be checked against the current MIGs and Conformance Suites prior to use.

## 3. CONTEXT

The SBR Reference Client is one part of the SBR SDK provided to assist Software Developers in adopting the SBR solution within their own software applications.

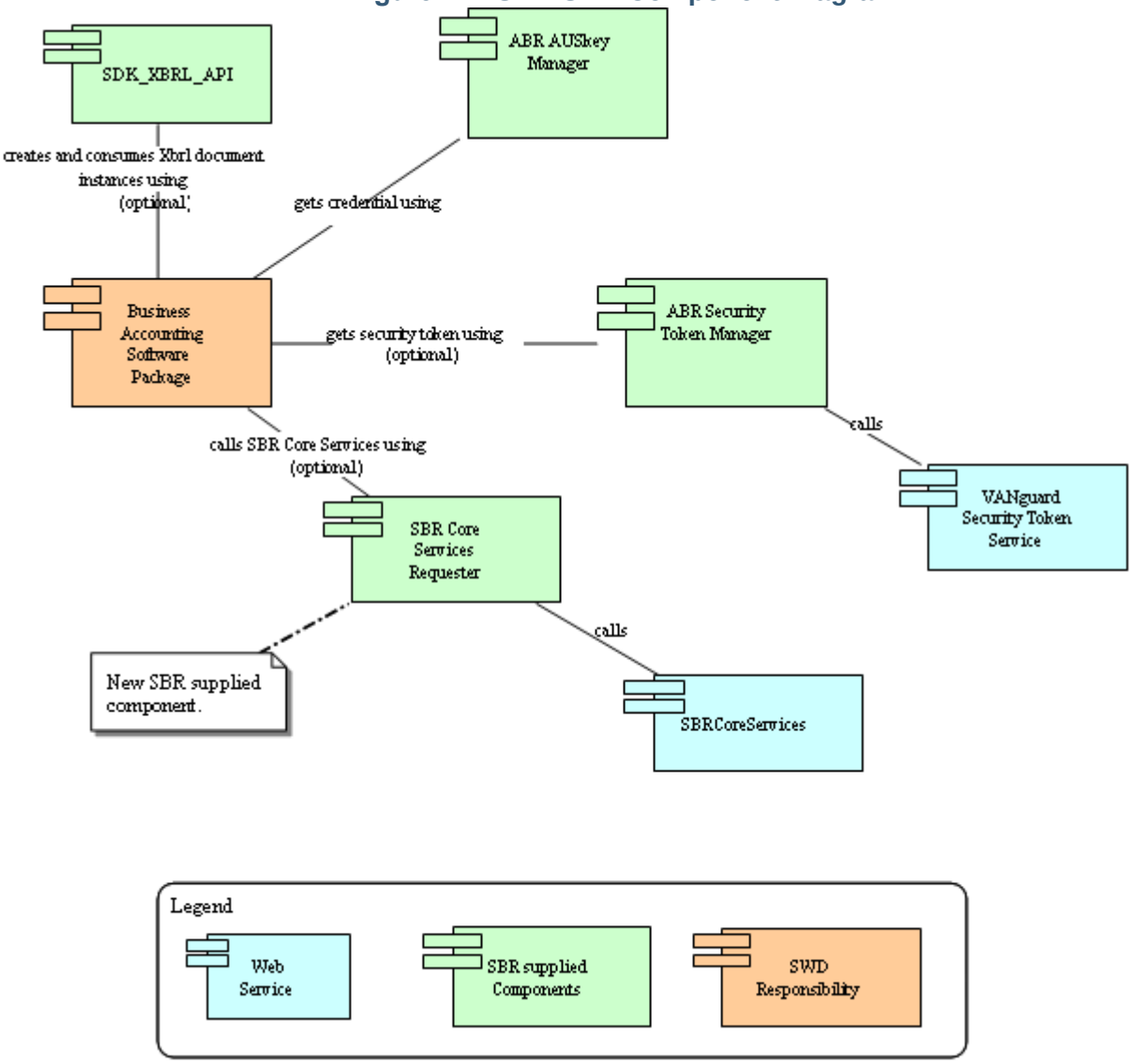
It provides a sample User Interface demonstrating how a client application can use SBR supplied software components to interact with SBR Core Services.

## 3.1. COMPONENT DIAGRAM

The following component diagram shows the high level interactions between a client application (Business Accounting Software Package) and the other components[1] which form the SBR solution.

The SBR Reference Client is a sample application which simulates a Business Accounting Software Package for the purposes of demonstrating use of the other components, with the exception of the SDK XBRL API.

**Figure 1: SBR SDK Component Diagram**

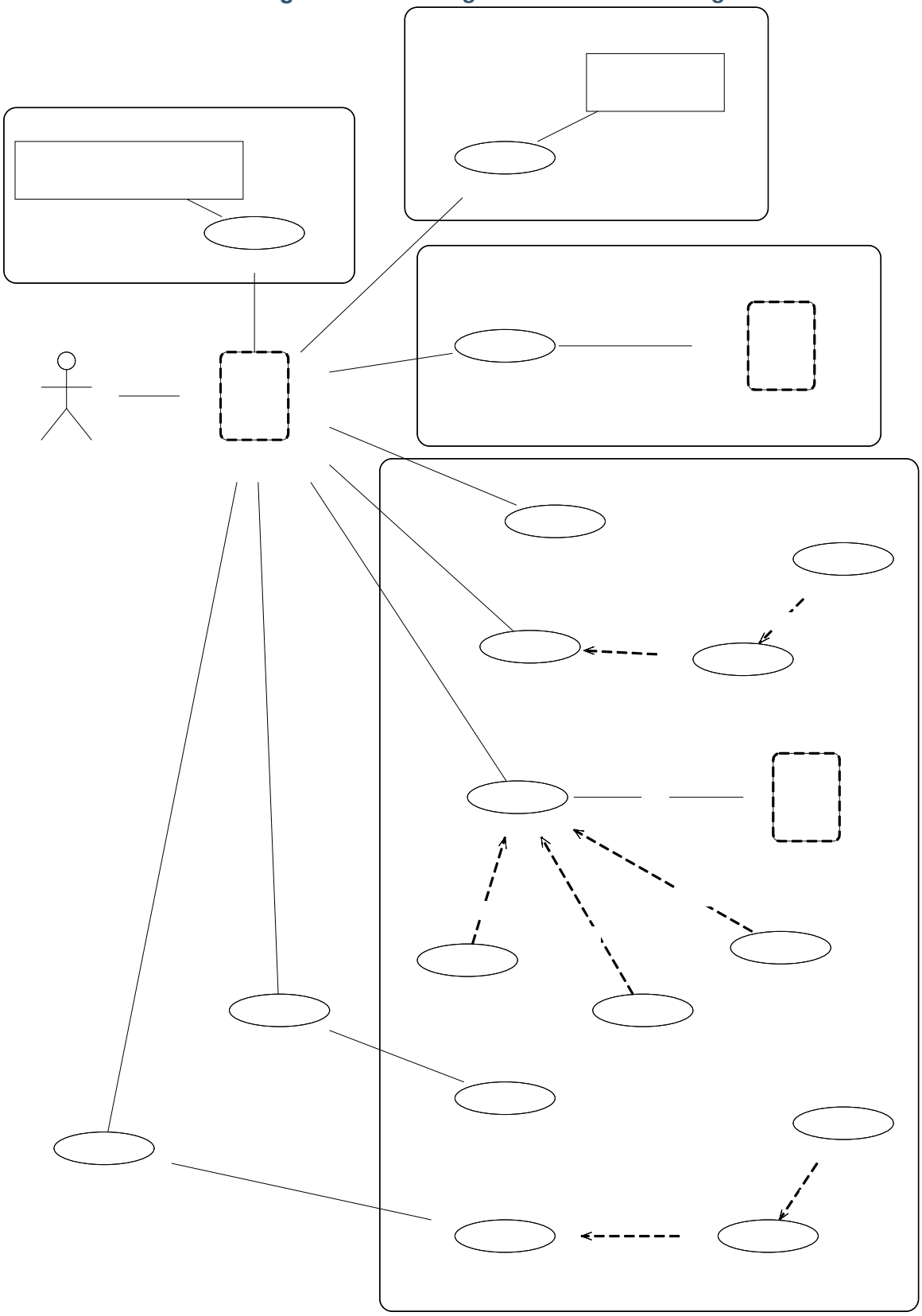



## 3.2. USE CASE DIAGRAM

The following use case diagram shows the primary functionality and scope of each of the components in the overall SBR solution.

[1] The SBR program is reviewing the naming of the components currently included in the Security Token Manager package. This document refers to the components using the new names (ABR AUSkey Manager and ABR Security Token Manager) and includes the previous names in brackets (ABR Keystore API and Security token Manager Request API respectively).

**Figure 2: SDK High Level Use Case Diagram**



# 4. DEPENDENCIES

## 4.1. JAVA JDK 6 AND ECLIPSE IDE

The SBR Reference Client has been built and tested using Java Development Kit (JDK) 6 and developed in the Eclipse Integrated Development Environment (IDE). It is also compatible with the NetBeans IDE.

## 4.2. ABR AUSKEY MANAGER (JAVA ABR.SECURITYTOKENMANAGER KEYSTORE API)

The SBR Reference Client uses the ABR AUSkey manager (ABR.SecurityTokenManager API) to facilitate the Authentication requirements for communicating with Core Services.

This SDK incorporates functionality to access client credentials in an AUSkey store and use the credential from the Keystore to request a token from the VANguard Security Token Service.

Refer to the **SBR Authentication Client SDK Developer Guide** document for further details on the ABR AUSkey Manager and its dependencies.

## 4.3. AUSKEY CREDENTIAL

In order to interact with the VANguard Security Token Service (STS) a valid AUSkey credential stored in an ABR Keystore is required.

The SBR Reference Client includes a test Keystore with a single credential which can be used for simple Message Ping testing.

This credential is stored under the Keystore folder of the SBR Java Client project and is a copy of the one distributed with the Identity Management SDK.

For testing beyond the simple Message Ping a set of test credentials should be obtained by registering with the SBR Program.

## 4.4. SBR CORE SERVICES REQUESTER COMPONENT

The SBR Core Services Requester component provides functionality to submit a request which implements the authentication and security required to connect to SBR Core Services.

Use of the SBR Core Services Requester component is demonstrated in the SBR Reference Client ClientBean, ResponseBean and SOAPFaultBean classes.

Refer to the **SBR Core Services Requester User Guide** for further details on its use and configuration.

# 5. COMPONENT SUMMARY

This section briefly describes each component and where its use is demonstrated within the SBR Reference Client.

### 5.1.1. ABR AUSkey Manager (ABR.KeyStore Keystore API)

The ABR AUSkey Manager (ABR.KeyStore Keystore API) provides functionality for managing and accessing the business client's SBR credential(s).

Use of the AUSkey Manager API to access credentials stored in a Keystore is demonstrated in the SBR Reference Client ClientBean and SecToken class files.

Refer to the **SBR Authentication Java Client SDK Developer Guide** for further details on its use and configuration.

### 5.1.2. ABR Security Token Manager (ABR.SecurityTokenManager API)

The ABR Security Token Manager (ABR.SecurityTokenManager API) provides functionality for accessing the VANguard Security Token Service to obtain Proof Tokens and Assertions required to access Core Services.

Use of the Security Token Manager to request a security token to connect to Core Services is demonstrated in the SBR Reference Client ClientBean and SecToken class file.

Refer to the **SBR SDK User Guide Java Identity Management Client** for further details on its use and configuration.

### 5.1.3. SBR Core Services Requester component

The SBR Core Services Requester component provides functionality to submit a request which implements the authentication and security required to connect to SBR Core Services.

Use of the SBR Core Services Requester component is demonstrated in the SBR Reference Client ClientBean.java file.

Refer to the **SBR Core Services Requester User Guide** for further details on its use and configuration.

### 5.1.4. SDK XBRL API

The SDK XBRL API provides functionality for creating, validating, and consuming XBRL business documents which contain the business information required for an interaction with an Agency.

Its use is not demonstrated within the SBR Reference Client.

Refer to the **SDK XBRL API User's Guide** for further details on the SDK XBRL API.

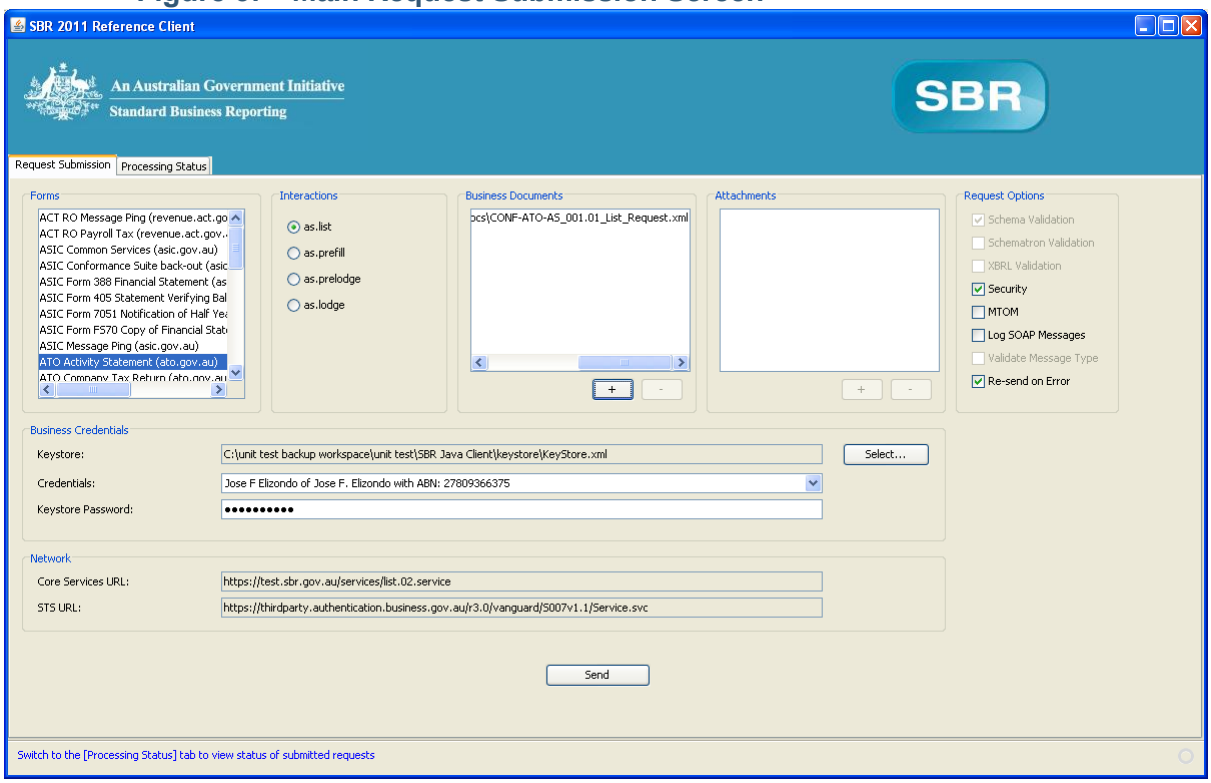
## 6. PARTS OF THE APPLICATION

## 6.1. SBR WINDOWS CLIENT

This is a Java client application which allows various options to be selected controlling which parts of the example to exercise.

### *6.1.1. User Interface*

**Figure 3: Main Request Submission Screen**




**SBRJavaClientView.java**

Contains the GUI definition.

The SBR Reference Client user interface has two tabs, one for submission of requests, and one to view progress of submitted requests and response details.

Note that the User Interface is not designed to be particularly usable or “pretty” as the main intent of the SBR Reference Client is to demonstrate the use of the components which form the SBR SDK.

**SBRJavaClientApp.java**

On startup the main view model for the user interface is instantiated and associated as the context for the GUI.

**BusinessDocumentBean.java** holds information related to the business document information that form part of the Standard Document Business Header (SBDH).

**AttachmentBean.java** holds information related to attachments on a business document..

**SOAPFaultBean.java** holds information that is displayed on the Exceptions tab located under Processing Status.

### 6.1.2. Converters

Several converters are defined which are used in Bindings within the GUI.

**InsertParametersToStringConverter.java**

This is the main one of interest as it is specific to SBR messages.

This provides an example of how a MessageEventItem which has parameters could be processed to substitute the parameter values into the short and long description text.

**XMLGregorianCalendarToDateConverter.java**

Provides an example of how to convert UTC dates to local time. The WIG states that all dates should be in UTC format, so any applications displaying dates should convert the dates to local time for display purposes.

**SBRCodeToStringConverter.java**

Provides a String conversion of the SBRCode data type.

### 6.1.3. Keystore

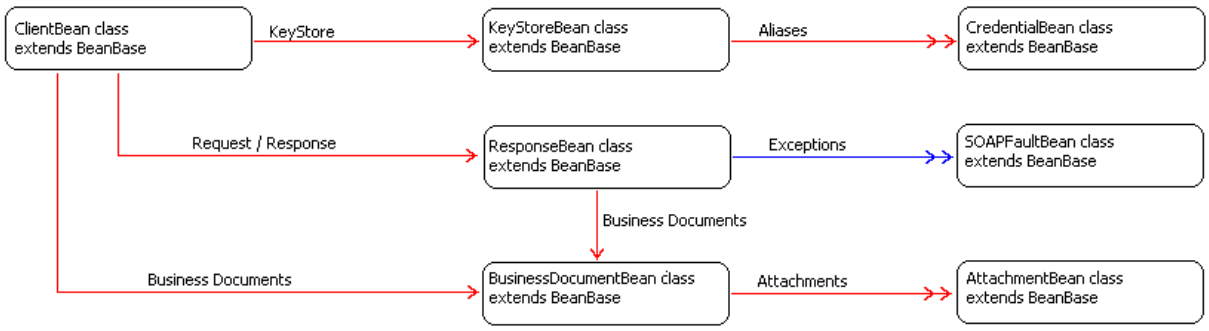
This folder contains a default Keystore and associated preferences file which can be used for simple message.ping tests.

### 6.1.4. Beans

The Beans provide a layer of abstraction between the GUI and the underlying data model. In the SBR Reference Client the majority of the code is contained in the various beans.

#### 6.1.4.1. ClientBean.java

This is the primary model used for the SBR Reference Client GUI. It contains all of the properties required by the user interface and controls the processing of requests and their responses.




#### 6.1.4.2. KeyStoreBean.java

This model encapsulates the functionality around accessing the Authentication Keystore.

The main underlying models are the AbrKeystore class from the Authentication API and a collection of ABR Credentials created from the set of credentials in the selected Keystore.

It allows the user to select a Keystore, creates a collection of credentials from the credentials in that Keystore and stores the Keystore instance for later use when a credential from the keystore is required for message processing.

Rather than using the default keystore the SBR Reference Client allows the user to select a Keystore to allow initial message.ping testing using the provided credential and later business function testing using additional testing credentials obtained by registering with the SBR Program.

#### 6.1.4.3. CredentialBean.java

This represents a single user credential from an Authentication Keystore.

It allows for selection of the credential and provides a string which is used in displaying the credential details in a dropdown list.

Note that the SBR Reference Client only demonstrates use of valid User credentials; it does not demonstrate use of Device credentials.

#### 6.1.4.4. BusinessDocumentBean.java

This model represents a business document.

Both the request and response messages for SBR Core services use the same format for Standard Business Document Message (SBDM), so this model is used in both setting up requests and displaying responses.

The underlying models are the business document information from the Standard Business Document Header (SBDH), the business document instance from the Standard Business Document Body (SBDB) and a collection of attachments as AttachmentBeans.

It provides functions for adding and removing attachments when setting up a request and viewing and saving business documents when viewing responses. This is implemented in the SBRJavaClientView.java class in conjunction with the BusinessDocumentBean.java class.

Viewing a document saves it to the system temporary path and displays it using whatever the associated default application is for viewing xml files.

#### *6.1.4.5. AttachmentBean.java*

This model represents an attachment to a business document and is again used in both setting up requests and displaying responses.

The underlying models are the attachment information from the Standard Business Document Header (SBDH) and the attachment instance from the Standard Business Document Body (SBDB).

It has a selection status, a display name used in list views and provides functions for viewing and saving attachments when viewing responses.

Viewing an attachment saves to the system temporary path and displays using whatever the associated default application is for files of the attachment's extension type.

#### *6.1.4.6. ResponseBean.java*

This model represents the Request/Response pair containing the request SBDM and the response SBDM or an Exception depending on the success or otherwise of the service call.

The underlying models are a Request SBDM instance and a Response SBDM instance or an Exception.

#### *6.1.4.7. SOAPFaultBean*

This model represents the SOAP Fault or error received from a form submission to core.

#### *6.1.4.8. BeanBase.java*

This is the base class for all of the Bean classes in the design pattern.

For more information about the source code and the implementation of the classes refer to the **Java Documentation** auto-generated from the building of the two projects.

# 7. USING THE REFERENCE CLIENT

## 7.1. CONFIGURATION SETTINGS

If you change any of the default settings it is recommended that you rebuild the SBR Reference Client using the ANT script build.xml file.

#### 7.1.1.1. Organisation Name

The organisation name by default is set to no value. If the default value is used, the SBR Reference Client will fail to run, therefore it is necessary to configure it in the file OrganisationNameCheck.properties before using the SBR Reference Client. This file is located in the au.gov.sbr.core.client.gui package.

#### 7.1.1.2. Resend time periods and endpoint settings.

The default resend time periods are 10 minutes for the initial resend and an additional 5 minutes each attempt after that. The default endpoints are set to the External Vendor Testing Environment (EVTE). In order to change these settings you can modify the ClientBean.properties file.

#### 7.1.1.3. Form properties

All properties for forms are located in the au.gov.sbr.core.forms package. They are used by the Form classes to extract message types, the agency domain and UI display.

#### 7.1.1.4. SBR Core Services Requester

For settings related to the SBR Core Services Requester refer to its documentation for further information.

#### 7.1.1.5. ABR Security Token Manager settings

For settings related to the Security Token Manager refer to its documentation for further information.

## 7.2. SETTING UP THE PROJECTS

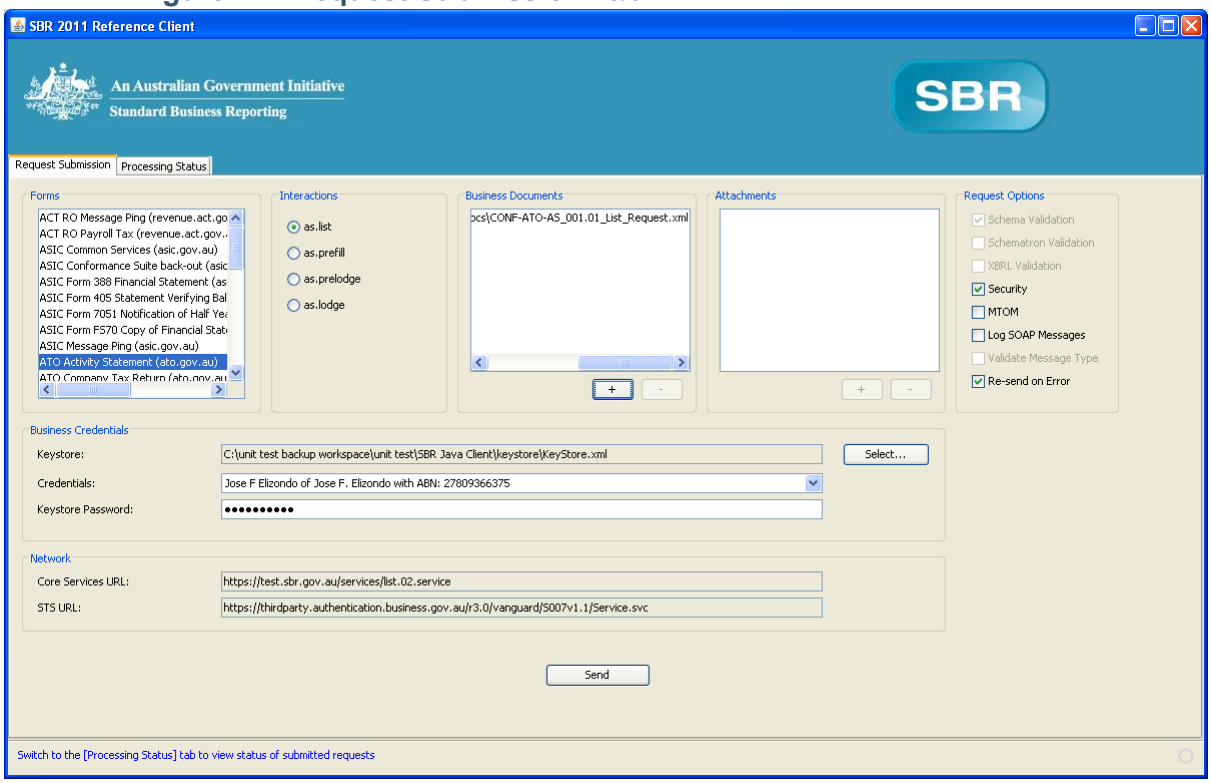
The download file is simply a zipped folder containing Java projects, so it can be unzipped and the “SBR Java Client” folder can be put in any location and imported into Eclipse IDE or Eclipse compatible IDE.

## 7.3. RUNNING THE CLIENT

The client is set up to run in the IDE, and Debug mode can be utilised to allow breakpoints to be set so that the code can be inspected as it runs. So for this purpose it’s best to run it from within the IDE.

## 7.4. REQUEST SUBMISSION TAB

**Figure 4: Request Submission Tab**




### 7.4.1. Forms

The Forms list shows the list of forms currently defined for use in the SBR Reference Client.

Please read **2.3 Forms defined in the Reference Client** for caveats related to the forms included with the reference client.

As of version 2.0.0 message.ping interactions are now defined as a form type rather than as a request option. This allows more flexibility to configure which core service options are available for each Agency.

### 7.4.2. Interactions

Shows the list of interactions defined for the selected form, select the one to be run from the radio buttons group.

### 7.4.3. Business Documents

A request to an agency generally includes one or more Business Documents in XBRL format. The current version of the SBR Reference Client does not demonstrate creation or validation of XBRL documents, so these should already have been created and available as files.

Press the [+] button to add a new document.

Press the [-] button to remove the selected document.

Removing a document will also remove any Attachments associated with that document.

### 7.4.4. Attachments

Some interactions with Agencies require attachments to be associated with Business Documents. At this point in time only ASIC accept attachments as part of their message interactions.

Press the [+] button to add an Attachment to the selected Business Document.

Press the [-] button to remove the selected Attachment.

### 7.4.5. Request Options

Request Options sets options which can be applied to the outgoing requests such as schema validation, signing of messages, use of MTOM for message attachments and logging the request and response messages.

**Schema Validation** turns on and off the schema validation of incoming replies against the basic WSDL schemas. This option currently cannot be turned off.

**Security** turns on and off the signing of the messages being sent. This can be turned off when using a local mock service or using the set of “no security” services which allows for testing of requests without requiring authenticated messages.

**MTOM** is used when attachments are to be sent to or received from agencies. At this point in time ASIC is the only Agency supporting attachments, so this is only required when calling ASIC services which require attachments. Refer to the MIGs for further information on which messages support attachments. Note that if a response is expected to include an attachment then the request should be sent using the MTOM binding even if there are no attachments in the request, otherwise the response will not be able to be parsed.

**Log SOAP Messages** turns on and off request and response SOAP message logging. The SOAP messages are printed out to the IDE console if the application is run from the IDE. In order to view the message logging when running from the jar file, ensure that the jar file is executed from the command line prompt.

**Validate Message Type** reserved for future expansion. This option currently cannot be turned on.

**Re-send on Error** turns on and off the ability to queue and resend a message should an exception or SOAP fault occur that is not critical. (e.g. service unavailable), Queuing times can be changed by modifying the ClientBean.properties file. The default times are: Re-Queue on 1st attempt 10 minutes and every attempt afterwards 5 minutes.

### 7.4.6. Business Credentials

The SBR Reference Client is set up to use the Keystore provided with the Identity Management SDK which contains a single credential for Jose Elizondo.

This credential is only valid for simple message.ping requests to agencies.

If additional tests are required an additional credential Keystore must be requested by registering with the SBR program.

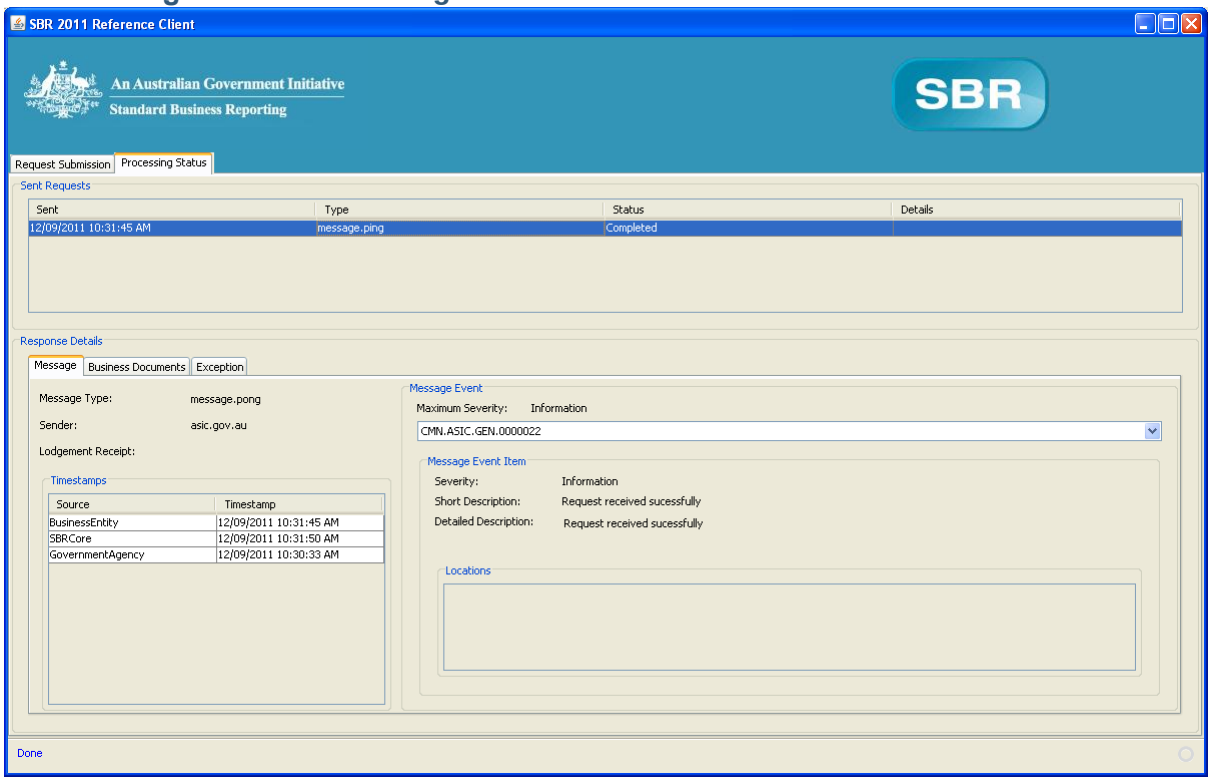
### 7.4.7. Send

The Send button is used to start the processing of a request to the selected agency using the business document, attachments, options and business credentials entered on the screen.

The Send button will only work once a form has been selected and if security is enabled, a Credential selected and password is entered. While a request is being processed it will show a processing indicator on the lower right hand corner and status text on the lower left hand corner.

## 7.5. PROCESSING STATUS TAB

**Figure 5: Processing Status Screen**




### 7.5.1. Sent Requests

This shows the set of Requests made in the current session. While a request is being processed it will show an ‘Outstanding’ status and on completion it will show either ‘Completed’ if a response was received, or ‘Exception’ if an exception occurred.

Note that a Completed message has not necessarily been successfully processed, it just means that a response was received from the service, this response may include an error message event from the agency.

### 7.5.2. Response Details

#### 7.5.2.1. Message Tab

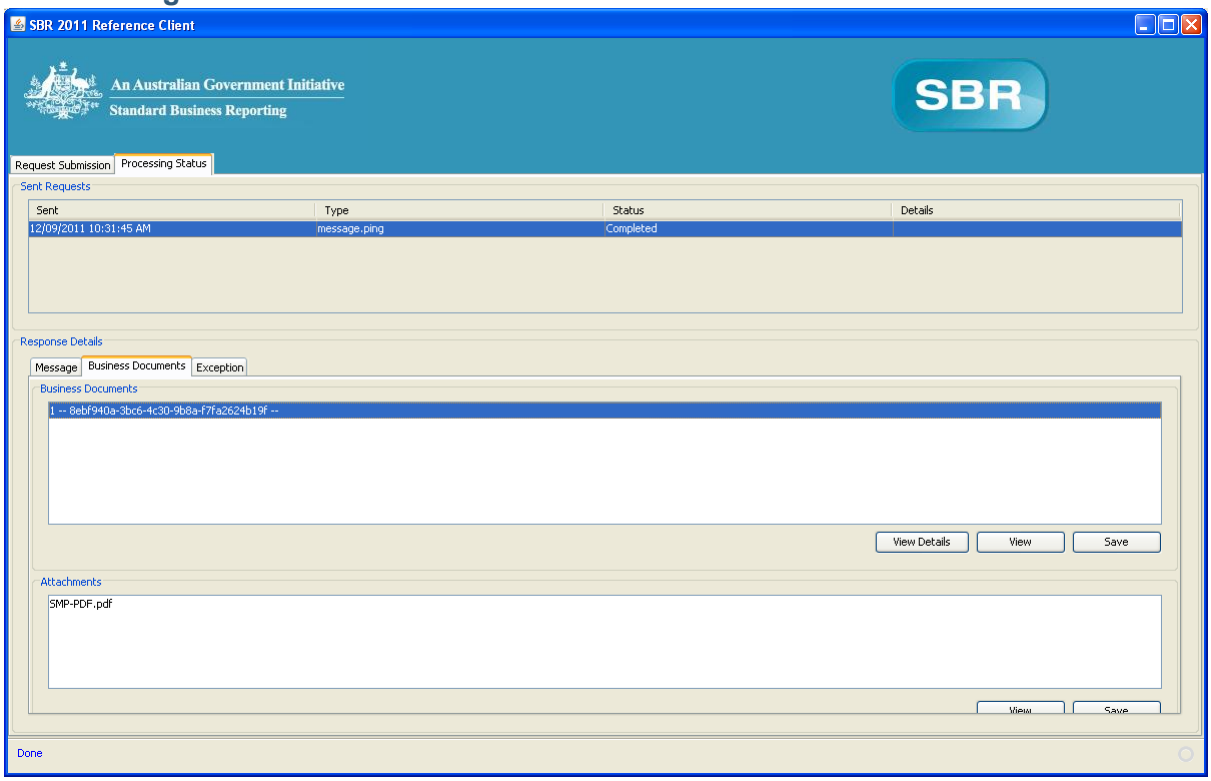
This tab shows the main details from the SBDH for the selected response from the Sent Requests.

The Message Event area shows the maximum error severity received and a drop down list of all Message Event Items in the response.

Details of the currently selected Message Event Item from the drop down list are displayed.

#### *7.5.2.2. Business Documents Tab*

**Figure 6: Business Documents Tab**



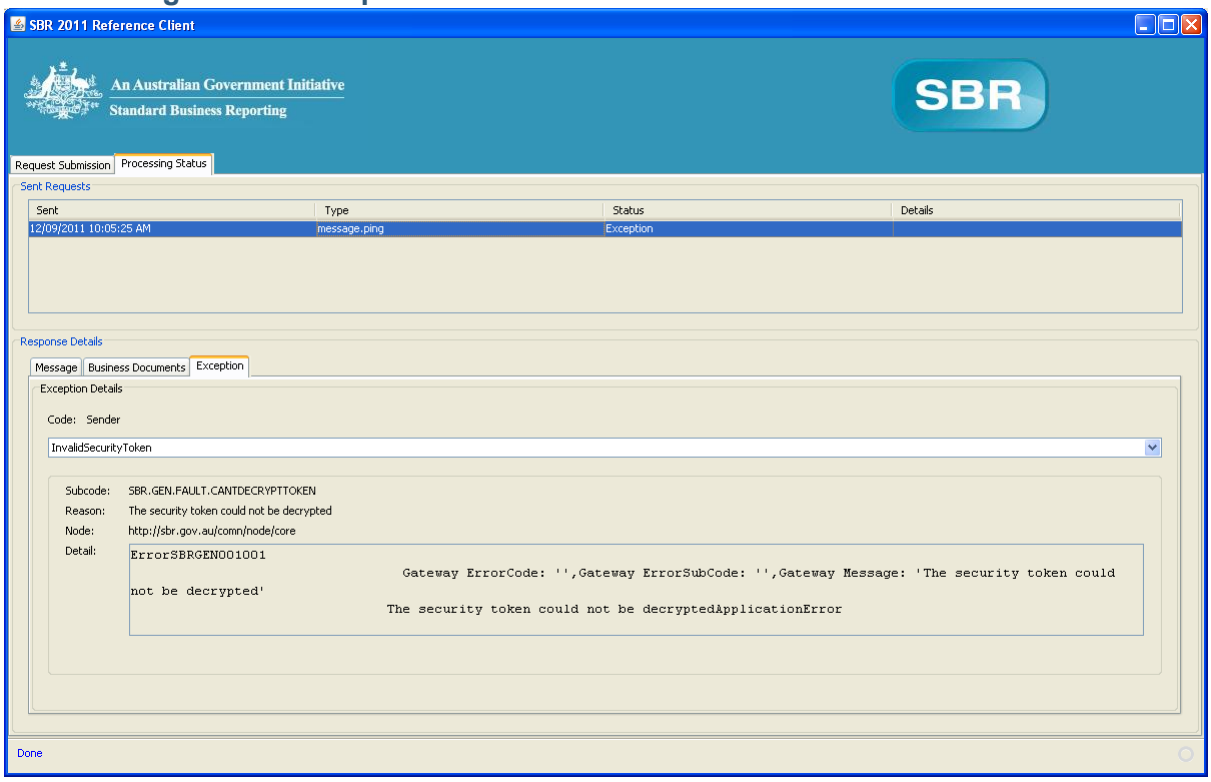

This tab shows the Business Documents received in the response and their associated Attachments.

The View Details button can be used to view the Business Document details from the SBDH.

If attachments were received the Attachments area will show the attachments for the selected Document.

#### 7.5.2.3. Exception Tab

**Figure 7: Exception Tab**




# 8. HINTS AND TIPS

## 8.1. MOCK SERVICE

Local mock services are useful for developing and unit testing clients without connecting to back end services. These are particularly useful for testing how a client responds to exception conditions which may not be able to be produced on demand.

There are a number of tools available which provide mocking capabilities through the use of a locally hosted endpoint and “canned” responses.

If the locally mocked service does not support SSL then configuration is required to modify the transport binding as the Core Services Requester component is designed to connect using HTTPS/SSL.

When using the SBR Reference Client to call a local mock service turn off the security option as otherwise an attempt will be made to request a security token which will fail.

## 8.2. TROUBLESHOOTING

### 8.2.1. Error SBR.GEN.GEN.4 Invalid message Type

**Scenario:** Trying to send a non Ping interaction to an Agency the response is an Error stating that message type [xxx] is not valid at location //sbdm:Message.Type.Text

**Likely Cause:** The Message Type defined in the form metadata is not correct. It may have been modified since the metadata was created.

**Remediation:** Check the Form properties file for the interaction being tried and check that the value specified for the MessageRequestType for the Service being called matches the value specified in the MIG for that Form.

Additional Information: By default the Reference Client uses the Form properties files contained in the SBR Java Client\src\au\gov\sbr\core\forms folder. Each form has a separate properties file containing xml which defines the Message Type values. Locate the value which is in error and correct it before running the Reference Client again.

At this point in time the Message Types include a version number, so they are likely to change with new versions of the Taxonomy making the Form properties provided with the Reference Client invalid.

### 8.2.2. A fault has been detected within the agency processing system

**Scenario:** A request (such as a Message Ping) is sent to an Agency (NSW) and the response is an Exception stating the above error.

**Likely Cause:** There are a number of reasons this might occur, however the most likely is that the Agency back end is not available.

**Remediation:** Check whether the Form properties include a valid value for the Message Type for the interaction being tried. If the message type defined is correct then it's possible that the Agency back end is not available – check with the SBR Service Desk to establish whether this is the case or not.

### 8.2.3. Unable to communicate with core because of PKIX error.

**Likely Cause** The SSL certificate is not recognised or trusted in the Java certificate store.

**Remediation** Use the Keytool utility that is provided with the Java runtime to import the server SSL certificate into the Java certificate store. Alternately, use the InstallCert utility that is freely available on the Internet to copy the server SSL certificate into a new trust file usually called JSSECACERTS. Once created, copy the file to the security folder of the Java runtime the SBR Reference Client uses.