



**ASIC**

Australian Securities & Investments Commission

# ***ASIC WEB SERVICES IMPLEMENTATION GUIDE (WIG)***

V3.0

**Purpose:** This document provides guidance to software developers for the implementation of ASIC register related web services offered by ASIC

**Date:** 28 February 2017

**Contact:** For further information or questions, contact ASIC at [webservices@asic.gov.au](mailto:webservices@asic.gov.au)

**DOCUMENT CHANGE CONTROL**

Version number	Date of issue	Author(s)	Brief description of change
0.1	20/07/2010	ASIC	Initial Draft
1.0	06/08/2010	ASIC	Draft reviewed for submission to external clients
1.1	22/02/2011	ASIC	Update structure of Header – add Attachments, Fees and Document No.
2.0	19/08/2011	ASIC	Update Business Document Header – and 3.3.1.11
2.1	06/02/2012	ASIC	Update Business Document Header – section 3.3.1.10 Add section 5.2
2.2	01/06/2013	ASIC	Add reference to VANguard & update STS to v1.2 Add index Remove 'Receipt' section as it is not used.
3.0	28/02/2017	ASIC	Update Business Document Header – section 3.3.1.11, 3.3.1.12, 3.3.1.13 and 3.3.1.14 Renamed document

**ACKNOWLEDGMENTS**

ASIC intends to be consistent with the documentation produced by other governmental projects. The template of this document is based on the Web Service Implementation Guide for Standard Business Reporting. The content of some sections might be similar or reproduce content from the Web Service Implementation Guide for Standard Business Reporting.

**DISCLAIMER**

This guide is current at the latest date shown in the Document Change Control above. ASIC may amend it at any time and without prior notice. ASIC is not responsible for the quality or merchantability of software developed on the basis of this guide, nor is it responsible for any application packages developed by third parties to access ASIC databases through the ASIC Web Services interface. ASIC does not support such software in any form.

**COPYRIGHT**

Copyright© Australian Securities and Investments Commission 2016 with all rights reserved

This document is the property of ASIC. No part of this document may be copied and used in other publications unless ASIC authorship is acknowledged.

# TABLE OF CONTENTS

<b>1. INTRODUCTION .....</b>	<b>6</b>
1.1. Purpose .....	6
1.2. Audience .....	6
1.3. XML specifications – please see <a href="http://www.w3.org">http://www.w3.org</a> for further informationContext .....	6
1.4. Note that testing that involves AUSKey authentication will also require use of a valid test AUSKey that is registered at the test instance of the Secure Token Service provided by VANguard. Glossary .....	8
1.5. Namespaces .....	9
<b>2. ASIC WEB SERVICE ARCHITECTURE .....</b>	<b>10</b>
2.1. Overview.....	10
2.2. Web Services .....	10
2.2.1. Services Offered .....	10
2.2.2. Web Service Standards .....	11
2.2.2.1. WS-Policy .....	12
2.2.3. Common Characteristics.....	12
2.3. Message Implementation Guides.....	13
<b>3. MESSAGE STRUCTURE.....</b>	<b>14</b>
3.1. Overview.....	14
3.2. SOAP Header.....	15
3.2.1. Security Element.....	15
3.3. SOAP Body .....	15
3.3.1. Business Document Header .....	15
3.3.1.1. Message Reference Number .....	16
3.3.1.2. ASIC Reference Number .....	17
3.3.1.3. Message Version .....	17
3.3.1.4. Sender ID.....	17
3.3.1.5. Sender Type .....	17
3.3.1.6. Software information .....	17
3.3.1.7. Message Timestamp.....	17
3.3.1.8. Message Event .....	18
3.3.1.9. Result.....	18
3.3.1.10. Attachments .....	19
3.3.1.11. Receipt.....	19
3.3.1.12. ASIC Transaction Number.....	20
3.3.1.13. Transaction Reference Number.....	20
3.3.1.14. Process Mode .....	20
3.3.2. Business Document Body.....	21
3.4. SOAP Faults.....	21
3.5. Dates and Times .....	21
3.6. Timeout Values .....	21
3.7. Duplicate transactions .....	21
<b>4. ERROR MANAGEMENT.....</b>	<b>22</b>
4.1. Overview.....	22
4.2. High Level Categorisation of Error Conditions .....	22
4.3. Transport Exceptions .....	23
4.3.1. SOAP Processing Model .....	23
4.3.2. Use of SOAP Fault fields .....	23
4.3.2.1. Code Element .....	23
4.3.2.2. Subcode Element.....	23

4.3.2.3.	Reason Element .....	23
4.3.2.4.	Role Element .....	23
4.3.2.5.	Detail Element.....	24
4.3.3.	Exception Conditions .....	24
4.3.3.1.	Client software errors.....	24
4.3.3.2.	ASIC web services unavailability.....	26
<b>5.</b>	<b>SECURITY.....</b>	<b>27</b>
5.1.	AUSKEY Credential .....	27
5.1.1.	STS Request HTTPS Connection Endpoint .....	30
5.1.2.	Creating the STS Request.....	30
5.1.3.	Processing the STS Response.....	32
5.2.	UserName & Password authentication .....	34
5.3.	Secure Messaging.....	35
5.4.	Signature Structures.....	35
5.4.1.	Identity Token <saml2:EncryptedAssertion> .....	37
5.4.2.	Business Certificate <wsse:BinarySecurityToken> .....	38
5.4.3.	Document Signature <ds:Signature id="signed_request_id">.....	38
5.4.4.	Envelope Signature <ds:Signature> .....	38
<b>6.</b>	<b>TESTING.....</b>	<b>39</b>
6.1.	Overview.....	39
6.1.1.	Service End Points.....	39
6.2.	Network Connectivity Testing.....	39
6.3.	Message Connectivity Testing .....	39
6.3.1.	Overview .....	39
6.3.2.	ping .....	40
6.3.3.	ping reply .....	41
6.4.	Message end-to-end Interaction Testing.....	41
<b>7.</b>	<b>ASIC WEB SERVICES REGISTRATION .....</b>	<b>41</b>
<b>INDEX.....</b>		<b>42</b>

## TABLE OF FIGURES

Figure 1:	Business Names Reference Materials .....	7
Figure 2:	Business Names M2M High Level Solution Overview .....	10
Figure 3:	ASIC BN - Structure of SOAP Message with Attachments.....	14
Figure 4:	SOAP Fault indicating XML is not well formed.....	26
Figure 5:	SOAP Fault indicating service not supported by ASIC .....	26
Figure 6:	SOAP Fault indicating ASIC processing system is unavailable.....	27
Figure 7:	Token.Request Sample Message.....	31
Figure 8:	Token.Response .....	33
Figure 9:	Sample XML - Reply authentication error event.....	35
Figure 10:	Sample Security Header.....	37

## TABLE OF TABLES

TABLE 1:	NAMESPACE PREFIXES .....	9
TABLE 2:	RECOMMENDATIONS USED BY ASIC WEB SERVICES .....	12
TABLE 3:	SOAP FAULT SUBCODES FOR CLIENT SOFTWARE ERRORS.....	25
TABLE 4:	STS CLAIMS .....	29
TABLE 5:	WSDLs FOR PING SERVICE .....	40
TABLE 6:	SPECIFIC VALUES FOR PING.....	40
TABLE 7:	SPECIFIC VALUES FOR PING REPLY .....	41

## 1. INTRODUCTION

### 1.1. PURPOSE

The purpose of this document is to provide information that will assist developers in the implementation of calls to the web services offered by ASIC. The document specifies many of the aspects that are, or should be, common to all or multiple message implementation specifications. These common aspects are primarily the structure of the SOAP messages, the security component and some of the parameters and data structures used in operation requests and replies.

### 1.2. AUDIENCE

The audience for this document is software developers in agencies, organisations or companies that will be building web services that interact with ASIC into their products. Readers should be familiar with the following:

- Company Registration – please see <http://www.asic.gov.au/for-business/starting-a-company/> for further information.
- Business Names – please see <http://www.asic.gov.au/asic/ASIC.NSF/byHeadline/Business%20names> for further information.
- Web Services – please see [www.ws-i.org](http://www.ws-i.org) for further information.

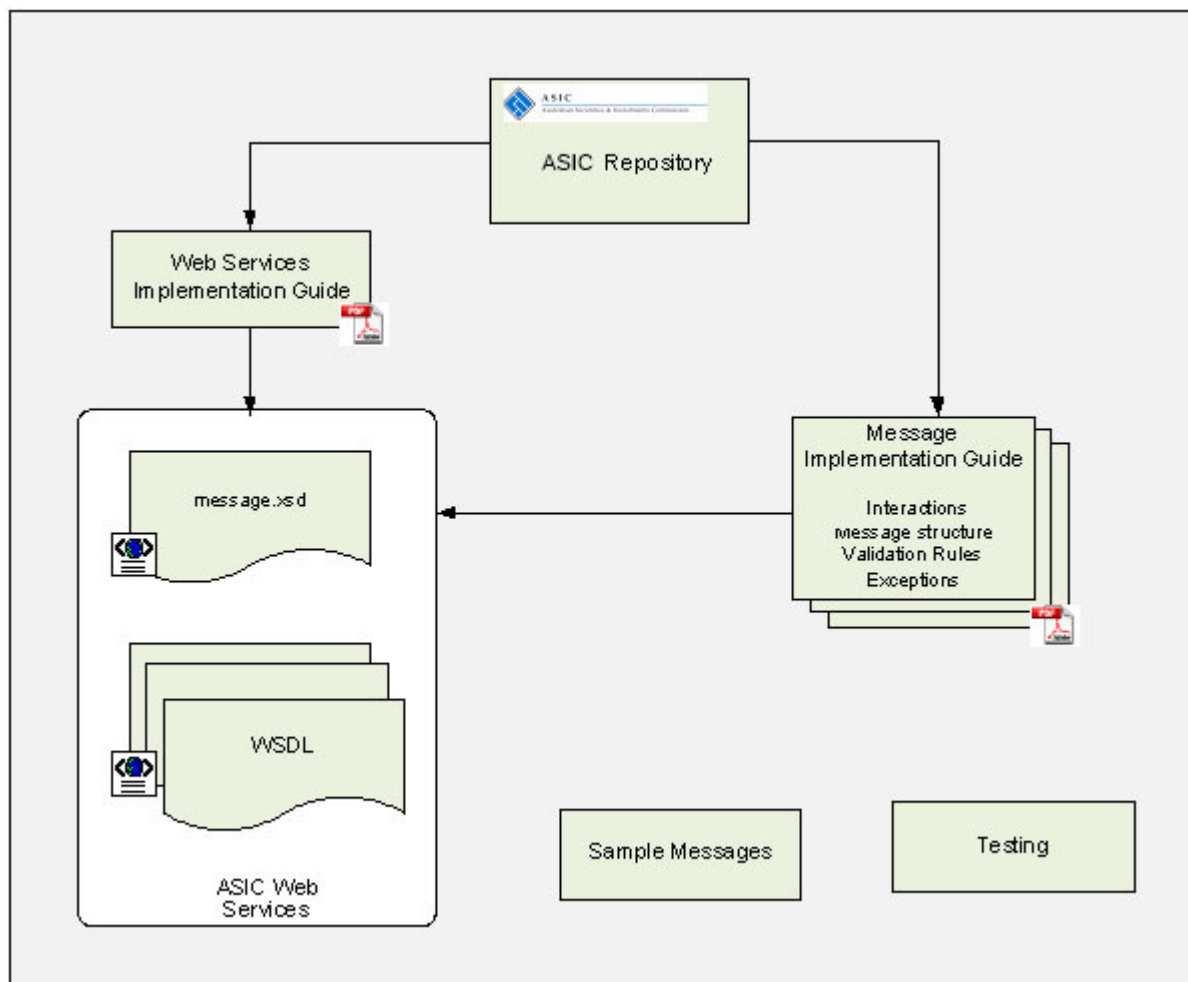
### 1.3. XML SPECIFICATIONS – PLEASE SEE [HTTP://WWW.W3.ORG](http://www.w3.org) FOR FURTHER INFORMATION CONTEXT

The ASIC web services program delivers a suite of documents and technical products to support software developers with the implementation of the web services available. These are illustrated in Figure 1.

All ASIC BN Web Services use XML documents within the SOAP envelope to query, to search, to update or to lodge applications.

Broadly speaking there are two groups of products:

- Implementation guides that provide the entry point for detailed information regarding how to implement the web services for the specific messages
- General support material such as test plans, test data, and other information that aims to facilitate efficient implementation.



**Figure 1: Business Names Reference Materials**

### **Web Service Implementation Guide (WIG)**

This document describes common technical components and services that are re-used by all business services. The common components include the standard web service protocols for message exchange, a standard business document message, a security token service, and a standardised approach to handling business error conditions and transport exceptions.

### **Web Services Description Language (WSDL) files**

The “Web Services Description Language” (WSDL) is the W3C standard syntax for the definition of web services. A WSDL describes the service, the information exchanged, and the technical protocols used for the exchange. ASIC provides specific web services together with a WSDL for each message. This collection also includes the XML Schemas for the messages.

## Message Implementation Guide (MIG)

The Message Implementation Guide (MIG) is the entry point for an implementer wishing to support ASIC service interactions. The MIG provides the business collaboration model, message business content for request and reply messages and all the business rules/validations and errors applicable to each request message.

## Testing

ASIC provides implementers with access to a supported test environment that can they use to verify their software against both the technical (web service) and business implementations before deploying them into production use.

### 1.4. NOTE THAT TESTING THAT INVOLVES AUSKEY AUTHENTICATION WILL ALSO REQUIRE USE OF A VALID TEST AUSKEY THAT IS REGISTERED AT THE TEST INSTANCE OF THE SECURE TOKEN SERVICE PROVIDED BY VANGUARD. GLOSSARY

For definition of the terminology and acronyms used within this document, please refer to the following list:

<b>ABR</b>	The Australian Business Register (ABR) is an extensive database of identity information provided by businesses when they register for an Australian business number (ABN).
<b>ABN</b>	Australian Business Number (ABN) is a single identifier for use in business dealings with the Australian Tax Office. Companies registered under the Corporations Act 2001 and business entities carrying on an enterprise in Australia are entitled to an ABN if they apply.
<b>AUSkey</b>	Online security credential that is a single secure sign-on for accessing a range of government online services
<b>BN</b>	Business Name – in the context of this document it is used as the name of ASIC's services relating to Business Names.
<b>Domain User</b>	Generic term used to define a user that will interact with BN using machine-to-machine services. Initially these are: ABR, agencies from States and Territories (S&T), ASIC Registered Agents and Information Brokers.
<b>HTTPS</b>	Hypertext Transfer Protocol Secure – a web protocol that provides encryption and secure connection
<b>ISM</b>	Australian Government Information Security Manual. A standard governing the security of government ICT systems. ( <a href="http://www.dsd.gov.au/infosec/ism/index.htm">http://www.dsd.gov.au/infosec/ism/index.htm</a> )
<b>M2M</b>	Machine to Machine – refers to automated inter-system data exchanges. Use of ASIC's BN web services is an example of an M2M interaction.
<b>SOAP</b>	Simple object access protocol, a computing standard defining how to format messages used by machine-to-machine interfaces.
<b>STS</b>	Secure Token Service – a service provided by VANguard to authenticate a device that wishes to access ASIC's BN web services based on their AUSkey.
<b>VANguard</b>	VANguard is an existing Whole-of-Government service delivered by the Department of Innovation, Industry, Science and Research that provides a range of secure e-



	authentication services.
<b>W3C</b>	World Wide Web Consortium (W3C) is the main international standards organization for the World Wide Web (abbreviated WWW or W3).
<b>WSDL</b>	Web Services Description Language

## 1.5. NAMESPACES

For conciseness, namespace definitions are not included in all examples. The appearance of the following namespace prefixes should be understood to refer to the corresponding namespaces from the table below.

<b>PREFIX</b>	<b>NAMESPACE</b>
env	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>
xmime	<a href="http://www.w3.org/2005/05/xmlmime">http://www.w3.org/2005/05/xmlmime</a>
xsi	<a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a>
wsse	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd</a>

**Table 1: Namespace Prefixes**

## 2. ASIC WEB SERVICE ARCHITECTURE

### 2.1. OVERVIEW

The following diagram illustrates, at a high level, the run time environment of the end-to-end ASIC machine-to-machine solution.

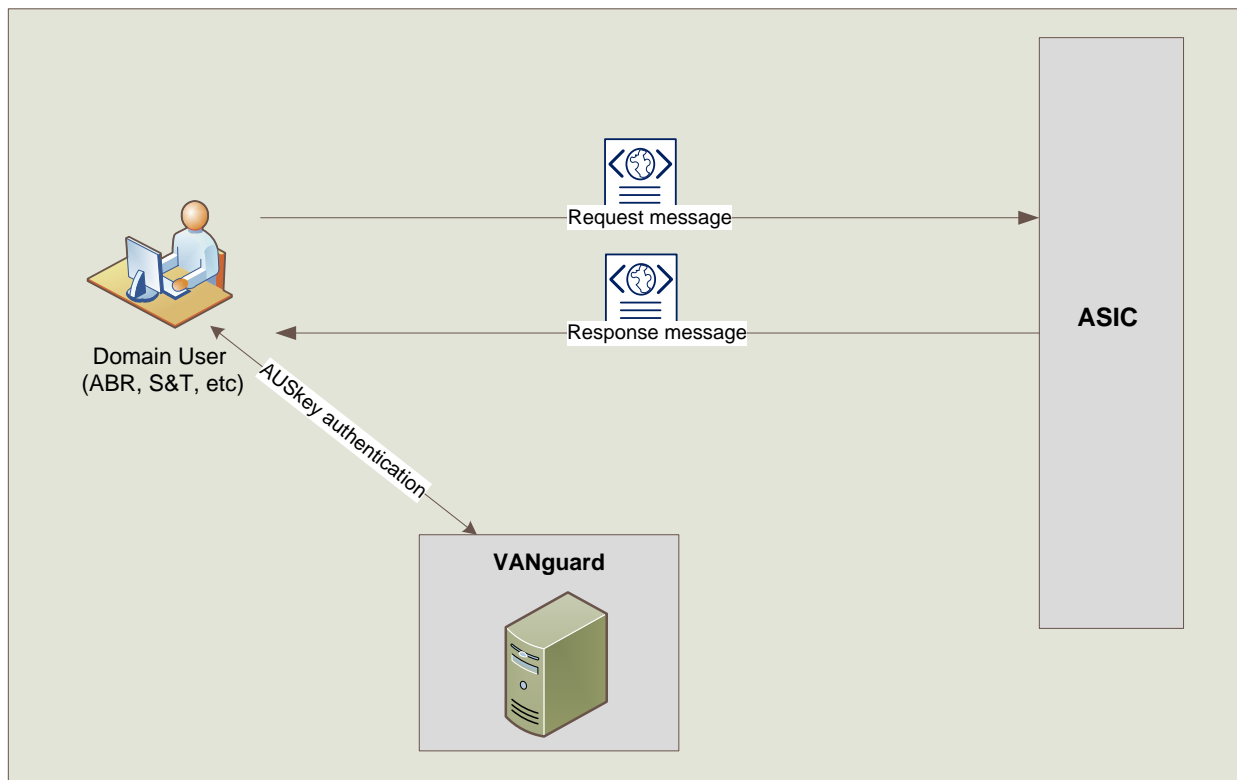


Figure 2: M2M High Level Solution Overview

### 2.2. WEB SERVICES

#### 2.2.1. Services Offered

ASIC exposes a number of Web Services that can be consumed by the software applications available to each group of users and the specific details for the messages are documented in the MIGs.

The collection of WSDLs, schemas and message samples are to be available from ASIC on request. It should be noted that the current message examples may not include security related headers.

The web services are authenticated using either the Secure Token Service mechanism provided by VANguard or the UserName & Password issued by ASIC.

The WSDL associated with the STS service, together with message samples will be provided to the interested parties on request, together with all other WSDLs. The WSDL for STS service is provided as a guide, with it being recommended that the actual WSDL be dynamically fetched from VANguard using the “?wsdl” convention.

All of the WSDLs, schemas and message samples can be obtained by emailing ASIC at [web.services@asic.gov.au](mailto:web.services@asic.gov.au). More information on AUSKey can be obtained from the ABR at <https://www.auskey.abr.gov.au/>.

More information on VANguard STS service can be obtained from <http://vanguard.business.gov.au> .

## 2.2.2. Web Service Standards

ASIC web services conform to the SOAP 1.2 recommendation provided by the World Wide Web consortium, and employ a range of related recommendations generically referred to as the “WS” suite. Table 3 below indicates the key recommendations employed by ASIC.

CATEGORY	APPLICABLE RECOMMENDATIONS
Transport	HTTP 1.1 HTTP over TLS (With constraints as per ACSI 33)
Messaging	RFC2392: Content-ID and Message-ID Uniform Resource Locators SOAP 1.2 MTOM 1.0
Description	WSDL 1.1 WSDL 1.1 Section 3.0 WSDL 1.1 Section 5.0 Namespaces in XML [World Wide Web Consortium 14-January-1999] XML 1.0 (Second Edition) XML Schema Part 1: Structures XML Schema Part 2: Datatypes
Security	TLS 1.0 HTTP over TLS RFC2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile WS-Policy1.5 Framework: W3C recommendations as of September 2007 WS-Security: SOAP Message Security 1.1 (WS-Security 2004) OASIS Standard Specification, 1 February 2006 Web Services Security: X.509 Certificate Token Profile 1.1 OASIS Standard Specification, 1 February 2006 Web Services Security: SAML Token Profile 1.1 OASIS Standard Specification, 1 February 2006 XML Signature Syntax and Processing Cryptographic Algorithms: Australian Government Information Security Manual, September 2012

**Table 2: Recommendations used by ASIC web services****2.2.2.1. WS-Policy**

WS-Policy provides a declarative framework in which to define many of the Quality of Service (QoS) parameters (security, compression, etc) associated with the ASIC web services. The vision is that these policy statements are portable across platforms and can be used by any platform and development framework to automatically configure support for the QoS requirements without developer intervention.

The set of recommendations supporting WS-Policy have only recently (in specification terms) been finalised by the W3C. Platforms have thus adopted interim versions of these specifications with the result that policy interoperability across platforms has yet to be achieved. In addition, not all platforms support the full set of policy assertions.

Below is the summary of ASIC's WS-Security policy for all web services:

- Messages are not encrypted since the web service uses SSL for transport layer encryption.
- Messages are signed using the WS-Policy endorsing supporting policy. Endorsing supporting policy requires that message digital signature generated by signing the elements defined in the security policy, should be signed again using the STS session key, thus generating a second signature.
- The primary digital signature is signed using an X509 certificate.
- The endorsing supporting digital signature is signed using an encrypted STS session token.
- The policy defines the trust claims required by the endorsing supporting token (STS session token) request.
- The policy defines the elements that are signed by both the primary and endorsing supporting digital signatures.

**2.2.3. Common Characteristics**

ASIC Web Services have the following common characteristics:

- Use of a request/reply style of interaction
  - With the exception of communication timeouts, any request will always receive a reply. All requests are initiated by the external parties' software (i.e. other governmental agencies, registered agents or information brokers).
- Synchronous calls
  - All web service calls are synchronous in nature. While most responses will be received within seconds, client software needs to be designed to cater for delays of the order of minutes.
- A standard security approach
  - All requests are secured in the same way across all services. Where security related information is returned in a response, a standard approach is also employed.
  - All supported authentication methods (AUSkey and username/password) are supported for all services
- A single mechanism to indicate the success of a request.

- All replies employ a standard mechanism to indicate the success of a request.
- If the request is successful, the business level reply data will be returned and optionally the URLs to PDF documents
- A standardised approach to the indication of failure conditions
  - Failures related to the transport of SOAP messages will be indicated using SOAP Faults, while business level errors are represented in a standard format within the reply message (see section 4 for more details).
- Request message sizes is limited to 50 MB
- Response message size is limited to 50 MB

### 2.3. MESSAGE IMPLEMENTATION GUIDES

The standards and common characteristics of the web services described above provide the building blocks from which the more complex collaborations needed to fulfil an interaction with ASIC registers. As far as possible, the web services and the business payloads they carry are loosely coupled so that additional messages can be added without requiring retesting of the web service infrastructure.

The way in which each web service is choreographed to fulfil a particular function is described within the **Message Implementation Guide (MIG)**.

There is a MIG for each group of users and in general it contains:

- The list and description of interactions specific to the governmental agency or business user
- The structure, content, rules for the business payloads in the request and the response messages
- The interaction specific values needed for the set of standard fields within the web service message structure.
- Validation rules and error messages for each service

Points in this document where the reader needs to refer to the MIG for message specific information are shown thus "Message Implementation Guide".

### 3. MESSAGE STRUCTURE

#### 3.1. OVERVIEW

ASIC web services employ a common message format shown in the diagram below.

All messages are carried over a one way HTTPS transport, and employ the SOAP 1.2 envelope structure. SOAP messages must employ UTF-8 or UTF-16 character encodings.

Details of the structures used within the SOAP Header and Body are described in subsequent sections. In the case of a discrepancy between this document and the WSDL schemas, the WSDL schemas take precedence and should be considered normative.

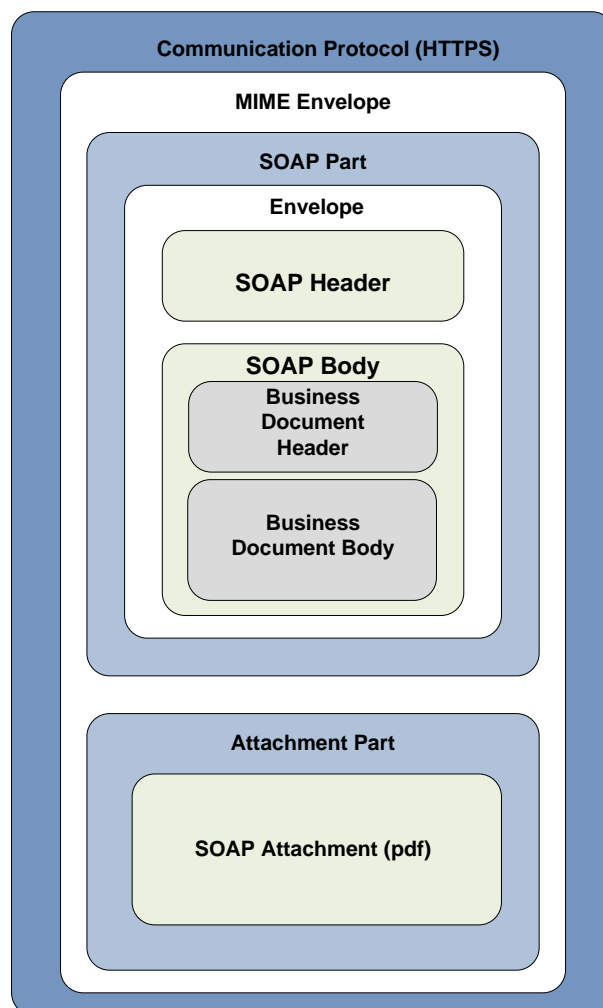


Figure 3: ASIC BN - Structure of SOAP Message with Attachments

## 3.2. SOAP HEADER

### 3.2.1. Security Element

Because ASIC has adopted the Web Service Security 1.1 recommendation, all security related information will be carried in the wsse:Security element within the SOAP header. Section 5 provides a detailed description of the security requirements and implementation.

## 3.3. SOAP BODY

ASIC follows the recommendations of the WS-I Organisation Basic Profile 1.0 and each SOAP body contains a single child element. The details for each particular message are available in the Message Implementation Guides.

As there is a high degree of communality in the message structures used for both requests and responses in ASIC web services, the SOAP payload has been logically divided into two segments as depicted in the Figure 3:

1. Business Document Header – contains metadata and common elements found in all messages
2. Business Document Body – contains business data specific to the message being transmitted

### 3.3.1. Business Document Header

The content of the Business Document Header carries the metadata required to facilitate the message exchange and the authorisation.

ELEMENT	PURPOSE	REQUEST	RESPONSE
messageType	Identify the type of message	Mandatory	Mandatory
messageReferenceNumber	Identifier for each particular message issued by the client	Mandatory	Mandatory
asicReferenceNumber	Identifier for reply messages issued by ASIC	Not provided	Optional
messageVersion	Identify the version of message	Mandatory	Mandatory
senderId	Identify the client who initiated the request. It is provided by ASIC when the client registers for used of ASIC Web Services.	Mandatory	Mandatory
senderType	Identify the domain users and specified by ASIC	Mandatory	Mandatory

softwareInformation	Identify the client's software	Optional	Not provided
messageTimestamps	Identify the date and time when the message was generated	Optional	Optional
messageEvents	Provide information for the client on the errors detected as a result of the business rules validation performed by ASIC	Not provided	Mandatory
result	Provide information on the result	Not provided	Mandatory
attachments	Provide information regarding the attachments returned by ASIC in the reply message	Not provided	Optional
receipt	Provide information regarding the fees	Not provided	Optional
asicTransactionNumber	ASIC transaction number assigned to a lodgement	Optional	Optional
transactionReferenceNumber	A unique transaction reference provided by the client. For some web services, this may be a mandatory field.	Optional	Optional
processMode	For some web services, this field may provide additional process request functionalities	Optional	Optional

### 3.3.1.1. Message Reference Number

The generator of the request message must include a unique reference identifier that will facilitate the audit trail and tracking of the messages. ASIC will return the same reference number in the response message.

ASIC will use this value for identifying duplicate transactions for messages that update data in the ASIC register - more details in section 3.7.



### 3.3.1.2. ASIC Reference Number

ASIC returns a unique reference number that may be used for querying the final status of a transaction.

### 3.3.1.3. Message Version

Each message will be versioned and the requestor must use the correct version at that point in time, as specified in the documentation provided. If in the future there is a need to change the versions of the messages, ASIC will support simultaneously the two versions of the message for a limited period of time to allow software developers to update their interfaces.

### 3.3.1.4. Sender ID

The 'senderID' element in the request message must be included only in the request messages.

### 3.3.1.5. Sender Type

The 'senderType' must be provided in the request messages.

### 3.3.1.6. Software information

The software information can be provided by the requestor and will be used for tracking and identifying issues within specific software products or versions.

The 'softwareInformation' contains the elements as indicated in the table:

ELEMENT	PURPOSE	Optionality
organisationalName	Name of the company that has build the software used for ASIC web service interactions	Mandatory
productName	Name of the product used for ASIC web service interactions	Mandatory
productVersion	Version of the product used for ASIC web service interactions	Mandatory

### 3.3.1.7. Message Timestamp

The generator of the message must include a timestamp to indicate the date & time when the message was created. In addition to the date/time value, a timestamp includes an indication of the entity generating the timestamp. Information regarding the format of the date/time value is found in the section 3.5.

ELEMENT	PURPOSE	Optionality
timestamp	Date and time when the message was generated	Mandatory
source	Entity adding the timestamp. The requestor must use the value "sender" ASIC must use the value "ASIC"	Mandatory

### 3.3.1.8. Message Event

In cases where messages fail business rules validations performed by ASIC's back-end systems, the reply will return one or many 'messageEvent' as part of the Business Document Header.

The messageEvent must contain the elements as indicated in the table.

ELEMENT	PURPOSE	Optionality
errorCode	A code to uniquely identify the condition that has occurred	Mandatory
severityCode	Items will be categorised by severity, with the available options: <ul style="list-style-type: none"> <li>Information</li> <li>Warning</li> <li>Error</li> </ul>	Mandatory
description	Descriptions on an item are intended to provide human readable text describing the error that has occurred.	Mandatory
locationPath	The location path field is included in the location to indicate, via an XPath expression, the element in the incoming XML document to which the event item refers.	Optional
details	Details regarding the error	Optional

ASIC intends to supply lists of errors applicable for each request message. These will be documented in the Message Implementation Guides.

### 3.3.1.9. Result

In order that every message exchange has an explicit indication of its result, every response to a service request must include one 'result' as part of the Business Document Header.

ELEMENT	PURPOSE	Optionality
accepted	'True' indicates that the request message has been accepted	Optional
rejected	'True' indicates that the request message has been rejected	Optional
delayed	'True' indicates that the message has not been fully processed by ASIC downstream internal systems	Optional
fault	'True' indicates that the downstream ASIC systems may be down and not able to process the request	Optional

### 3.3.1.10. Attachments

For some of the interactions ASIC will provide information rendered in PDF format that can be retrieved from an ASIC location. The appropriate metadata including the URL to the storage location will be indicated in this section. Detailed instructions regarding the type, URL, name and number of PDF documents are provided in any Message Implementation Guide that contains a message that requires attachments.

ELEMENT	PURPOSE	Optionality
documentType	The type of the attachment content	As specified in the MIG
binaryObject	The base64 encoded value of the attachment	Not applicable
URL	The URL for the location of the PDF documents	As specified in the MIG
documentNumber	The number assigned by ASIC for the document	As specified in the MIG
name	The name of the file	As specified in the MIG
description	A short description of the content	As specified in the MIG

### 3.3.1.11. Receipt

Fees information is present in ASIC replies where the request message triggers a fee, such as lodgement of an application for a new business name or requests for extracts (e.g. business name extract, person extract)

Seq No	ELEMENT	PURPOSE	Optionality
1	debtor	Contains elements related to the client	Mandatory
1.1	account	Contains elements related to the client's account such as type of entity, ledger used and the unique ASIC identifier (e.g. ACN, ARBN, RA Number) assigned to the client	Mandatory
1.1.1	asicType	Indicates the type of entity in ASIC register that the client is categorized as.	Mandatory
1.1.2	ledgerNumber	ASIC ledger number used for the type of entity	Mandatory
1.1.3	asicIdentifier	The value of the ASIC unique identifier assigned to the client such as ACN, ARBN, ARSN, Registered Agent Number, Business Name Number.	Mandatory
1.2	name	Name of the debtor	Mandatory

1.3	abn	ABN of the debtor	Optional
1.4	abnReference	ABN reference number of the debtor, if ABN not issued yet.	Optional
1.5	address	The address of the debtor	Mandatory
2	dateIssued	Date of the receipt	Mandatory
3	transaction	Details regarding the transactions that trigger the debt	Mandatory
3.1	reportingParty	Details regarding the party incurring the debt	Mandatory
3.2	date	Date of the transaction	Mandatory
3.3	descriptionLine1	Description of the item on the transaction	Mandatory
3.4	descriptionLine2	Description of the item on the transaction	Optional
3.5	reference	ASIC reference for the transaction	Mandatory
3.6	amount	Amount charged for the transaction	Mandatory
4	total	Total amount for all transactions	Mandatory
5	totalDue	Total amount due	Mandatory
6	dueDate	Due date	Optional

#### 3.3.1.12. ASIC Transaction Number

The 'asicTransactionNumber' is the unique identifier used for recording the documents lodged with ASIC. It is provided in the reply header for transactions that are updating ASIC register. The presence of this element will be indicated in the MIG.

#### 3.3.1.13. Transaction Reference Number

The 'transactionReferenceNumber' is a unique number provided by clients to ASIC. It is intended to be used by the client to track their transaction and related sub transactions.

#### 3.3.1.14. Process Mode

The 'processMode' is used by some web 'lodgement' services only. If used, it provides the option to validate the request data only and returns any errors found. If left blank or set to "PROCESS", it will default the message request as a "lodgement" request.

### **3.3.2. Business Document Body**

The Business Document Body will contain the business level data required to complete the business interaction.

The Message Implementation Guides (MIGs) will detail the structure, the rules and the validations for each message. For documentation purposes the schemas for request message and response message for each Business Document Body will be made available in the MIGs.

### **3.4. SOAP FAULTS**

SOAP faults will return the standard fields defined in the SOAP specification and they will not include detailed information on the format of "Detail" element.

### **3.5. DATES AND TIMES**

All dates and times are expressed in messages as per the standard XSD built-in "datetime" data type, as specified in <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/#dateTime> which is a subset of the ISO-8601 standard.

Timestamps must be expressed in Australian Eastern Standard Time (AEST) and should be provided with time to the second e.g.

2009-03-25T13:53:48

It should be noted that where date/time values are displayed to users, they will first need to be converted into the local time zone.

### **3.6. TIMEOUT VALUES**

There are a number of sources of delay between the issuing of a request by client software, and the delivery to the software of the resulting response. These include transmission delays between the client software and ASIC systems as well as processing delays at ASIC backend systems.

While most requests are expected to be processed within the order of 10 seconds, ASIC interface will allow up to 2 to 3 minutes for a request to be processed by the backend systems.

Developers should thus configure their products to use a timeout of somewhat more than 3 minutes, depending on the transmission delays to ASIC expected as a result of the capacity of their client's internet connections and the nature of the services being supported.

### **3.7. DUPLICATE TRANSACTIONS**

In general, the interactions available for ASIC web services can be divided into:

- query type messages – where the message is retrieving existing data in ASIC system and
- transaction type messages - where the request message is adding/updating data in ASIC registry

There may be occasions when business software may not receive the response message due to a number of failure modes (e.g. loss of internet connectivity) for which the logical action by requestor software should be to resubmit the request.

For transaction type messages, the resubmitted request must use the same value for the 'messageReferenceNumber' as the initial failed request. One of the following two scenarios may occur:

1. the initial message is already fully processed and the original response is returned or
2. the initial message was not received or processed by the ASIC system in which case the interaction will follow the typical interaction pattern and the requestor will receive a response message.

## 4. ERROR MANAGEMENT

### 4.1. OVERVIEW

This chapter details the approach to be taken to the handling of errors and exception conditions associated with the submission of requests to ASIC via Web Services.

Errors may result from a number of sources such as: authorisation error; user errors; client software problems; XML or message structure invalid, ASIC processing systems are not available; errors internal to ASIC systems and so on.

We assume that the user and client software errors should be detected and remediated by the client software providers. Thus, this document is not covering these types of errors.

### 4.2. HIGH LEVEL CATEGORISATION OF ERROR CONDITIONS

The error conditions and exception can be categorised into four broad areas:

1. Transport exceptions
2. Business rule errors

Errors associated with the physical transport of messages from businesses to agencies fall into the 'Transport exceptions' category. This area covers any problems related to ensuring SOAP messages are successfully passed from sender to ASIC and back again. This document completely describes the architecture to handle transport exceptions, as well as enumerating the conditions which fall within this category.

Having established an error free flow of SOAP messages, all remaining error conditions will be as a result of business activities. The term "business event" is used since some of the conditions flagged at this level are of an informational or warning nature, and won't necessarily represent an error condition per se.

Business rule errors will be communicated via the "messageEvent" structure within the SOAP Body - Business Document Header.

## 4.3. TRANSPORT EXCEPTIONS

### 4.3.1. SOAP Processing Model

ASIC web services has adopted the W3C SOAP 1.2 recommendation, together with an HTTP based transport, as the basis for the on-the-wire format of messages to be exchanged between client software and ASIC. This implies the solution will be consistent with the SOAP processing model, and will leverage the SOAP fault mechanism as the primary way in which transport or server related exceptions will be communicated.

It is recommended that the reader familiarise themselves with the SOAP specification prior to reading the remainder of this document.

The software generating requests to ASIC takes the role of the initial sender while ASIC's processing system takes the role of receiver.

Because ASIC web services use the Request/Response Message Exchange Pattern (MEP) for all exchanges between agencies and ASIC, a successful exchange will see a valid SOAP message generated by the initial sender, forwarded to ASIC actioned by the ASIC processing system, and a valid response message generated and returned to the initial sender.

### 4.3.2. Use of SOAP Fault fields

The SOAP 1.2 recommendation defines a number of standard fields within a SOAP Fault.

#### 4.3.2.1. Code Element

ASIC web services will conform to the SOAP 1.2 recommendation and use only the standard values it defines for the Code element.

As described in the recommendation, a code of "env:Sender" carries with it an expectation that the sender will change/correct the request, whereas "env:Receiver" implies a downstream problem with no change needed to the request.

#### 4.3.2.2. Subcode Element

ASIC web services will use subcodes in order to provide the necessary detail. More than one level of subcode may be necessary in order to establish the high level action required in each case. For example, subcodes defined by BN will use the namespace prefix "asic.bn", which will be mapped as per section 1.5.

#### 4.3.2.3. Reason Element

The reason element will describe the specifics of the particular error condition, and will thus reflect the finest granularity of subcode provided in the fault.

#### 4.3.2.4. Role Element

Role elements will not be included in SOAP faults.

#### 4.3.2.5. Detail Element

It is not intended that SOAP faults provided to client software in the production environment carry any additional information over and above the codes and reason elements already described. Thus they will be provided without a Detail element.

### 4.3.3. Exception Conditions

There are a range of conditions that may occur while exchanging messages with ASIC.

#### 4.3.3.1. Client software errors

All errors in this category result from a defect within the software used by business causing the generation of invalid requests. It is intended that the ASIC testing regime will allow the detection and rectification of this category of errors. However the possibility still exists that such conditions will occur at runtime, perhaps triggered by edge cases in data contents.

In line with the SOAP 1.2 recommendation, errors in this category will mostly be reported with a SOAP fault using a code value of "env:Sender". The only exceptions to this are conditions called out by the W3C recommendation itself such as "env:VersionMismatch".

It is expected that if errors in this category occur during the operation of a software package, the business user would be expected to contact their software provider for rectification advice.

The subcodes below will be used to indicate the specific error condition detected.

SUBCODE	REASON	DESCRIPTION/COMMENT
asic:FAULT.MALFORMEDXML	The request was not well formed XML.	The request is not well formed XML, as documented in the XML specification.
asic:FAULT.INVALIDXML	The request does not validate against the service XML Schema	The request does not validate against the XML Schema for the service, which is defined as part of the WSDL for the service.
asic:FAULT.TOOBIG	Request size limit exceeded	The NB limit on maximum request size is set to 50 MB
asic:FAULT.ATTACHMENTERROR	The request message attachment could not be processed.	The request message contained an attachment that could not be processed.
asic:FAULT.UNKNOWNSERVICE	Unknown service	A request has been made for a service not offered by the receiving party.
wsse:SecurityTokenUnavailable	No security token was provided	A VANguard supplied security token must be included in the request.
wsse:FailedCheck	The provided signatures or encryption were invalid	This error will occur as a result if <ul style="list-style-type: none"> <li>- the security token does not decrypt successfully</li> <li>- the session signature was not valid</li> </ul>



<p>wsse:InvalidSecurityToken</p>	<p>An invalid security token was provided</p>	<p>This error covers a number of circumstances related to the security token :-</p> <ul style="list-style-type: none"> <li>- The signature of the provided token was invalid</li> <li>- the token has expired</li> <li>- the token did not include a session key</li> <li>- the token did not include the required set of mandatory claims</li> <li>- the session key could not be decrypted</li> </ul> <p>A current VANguard supplied security token must be included unaltered in the request. The certificate presented to obtain this token should be used to sign</p>
<p>asic:FAULT.UNAVAILABLE</p>	<p>Unavailable service</p>	<p>A request has been made for a service that is currently unavailable.</p>

**Table 3: SOAP Fault Subcodes for client software errors**

The figures below show examples of the faults that will be generated as a result of client software errors. Line wraps within text are for presentational purposes only.

```
<env:Fault >
  <env:Code>
    <env:Value>env:Sender</env:Value>
    <env:Subcode>
      <env:Value>asic:FAULT.MALFORMEDXML</env:Value>
    </env:Subcode>
  </env:Code>
  <env:Reason>
    <env:Text xml:lang="en">The request was not well formed XML</env:Text>
  </env:Reason>
</env:Fault>
```

**Figure 4: SOAP Fault indicating XML is not well formed**

```
<env:Fault >
  <env:Code>
    <env:Value>env:Sender</env:Value>
    <env:Subcode>
      <env:Value>asic:FAULT.UNKNOWNSERVICE</env:Value>
    </env:Subcode>
  </env:Code>
  <env:Reason>
    <env:Text xml:lang="en">Unknown service</env:Text>
  </env:Reason>
</env:Fault>
```

**Figure 5: SOAP Fault indicating service not supported by ASIC**

```
<soapenv:Fault >
  <soapenv:Code>
    <soapenv:Value>env:Sender</soapenv:Value>
  </soapenv:Code>
  <soapenv:Reason>
    <soapenv:Text xml:lang="en">Unknown rule in CALL</soapenv:Text>
  </soapenv:Reason>
</soapenv:Fault>
```

**Figure 6: SOAP Fault indicating message version not supported by ASIC**

#### 4.3.3.2. ASIC web services unavailability

It is important that client software is aware of the “normality” of this error condition and take the necessary steps to resubmit the request at a later time. This may involve automatic queuing of the request for resubmission at a later time, or notification to the user that they should initiate the resubmission after a suitable delay. Where resubmission is automated, it is recommended that an increasing delay be added between resubmission attempts.

In some cases, the time at which the service will be available again is known. In this case, the reason text should contain the date and time (including time zone) after which the service is

expected to be available again. In addition, in order to allow automatic re-queuing of the request, the fault detail will contain the equivalent information in a machine consumable format (see example below). The presence of this information should be checked for, and where possible, used to re-queue the request. The human readable and machine consumable times may not align exactly, in order to allow the spreading of requests over time after the service resumes.

Regardless of the approach taken by the software, it is important that the indications provided to business users ensure they understand there is no need to contact either their software provider or ASIC. Software developers should adopt an optimistic approach to request submission, taking into account any information provided in regards to the date and time at which the service will be available again.

All the conditions resulting in unavailability of ASIC will be reported to client software with a code value of “env:Receiver” and a subcode of “asic:FAULT.UNAVAILABLE”

```
<env:Fault >
  <env:Code>
    <env:Value>env:Receiver</env:Value>
    <env:Subcode>
      <env:Value>asic:FAULT.UNAVAILABLE</env:Value>
      <env:Subcode>
        <env:Value>asic:FAULT.ASICNOTRESPONDING</env:Value>
      </env:Subcode>
    </env:Subcode>
  </env:Code>
  <env:Reason>
    <env:Text xml:lang="en">The connection timed out</env:Text>
  </env:Reason>
</env:Fault>
```

**Figure 7: SOAP Fault indicating ASIC processing system is unavailable**

## 5. SECURITY

This section will only describe the security aspects associated with the ‘message on the wire’. The Business Name services may be accessed using AUSKey credentials or using a combination of User Name & Password via basic HTTP authentication.

It is assumed that the domain user has already acquired:

- AUSKey credential from the Australian Business Register (ABR) and has installed it in their software package. Details of the registration and certificate issuing process are provided on the ABR website.

Or

- User Name and password issued by ASIC

### 5.1. AUSKEY CREDENTIAL

The principle security interactions are shown below.

- The business software presents their business certificate to the Security Token Service (STS), it is authenticated, and receives a session key and an encrypted token.

- The business software uses their certificate to sign the SOAP Body within each request, and incorporates this, together with the encrypted token from the STS into the WS-Security header.
- The STS token and the entire SOAP body are signed using the session key obtained from the STS. The complete SOAP message is then sent to ASIC
- ASIC identifies and authorises the domain users using data in the decrypted token, processes the business message and returns a response.

The STS interaction (Token.Request and Token.Response) is an implementation of the standard Web Service Protocol WS-Trust (<http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.html>)<sup>1</sup>.

The ASIC web services interaction (Service.Request & Service.Response) is an implementation of the standard Web Service WS-Security protocol (<http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>)

The security token is an implementation of the standard SAML 2.0 protocol (<http://saml.xml.org/saml-specifications>).

In the Token.Request message, the client passes a set of claims to the STS, with confidentiality being provided using SSL. To indicate the source of the request, a timestamp is signed using the business private key.

The Security Token Service maintains meta-data that relate to a business credential and will return a Token.Response that contains a set of assertions (the values associated with the claims), packaged as a security token. The STS also provides a session key with a lifetime of 30 minutes. That session key can be used for any number of secure interactions with ASIC web services until expiry.

Within a Token.Request, claims may be marked optional. If a claim is marked optional, the STS will return an assertion if it has a value for the claim and no assertion otherwise. If a claim is not marked optional and the STS does not have a value for the claim, this error condition will be flagged by the return of a SOAP fault rather than a Token.Response.

The table below lists the 16 claims that must be included in each request to the STS. It also shows, for each type of credential whether the resulting assertions must be present within a token included in a request to ASIC.

Given the STS behaviour in terms of optionality of claims, and in order to simplify the logic necessary to support both credential types, it is recommended that all 16 claims be requested as optional in each STS request, except for those in the table below where the claim is marked as mandatory for both credential types. An example is shown below in section 5.1.1.

CLAIM URI	DESCRIPTION	ABR_User	ABR_Device
<a href="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/abn">http://vanguard.ebusiness.gov.au/2008/06/identity/claims/abn</a>	Sender ABN	Mandatory	Mandatory
<a href="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/commonname">http://vanguard.ebusiness.gov.au/2008/06/identity/claims/commonname</a>	User full name (ABR_User) Server name (ABR_Device)	Mandatory	Mandatory

<sup>1</sup> Note – the examples may include sample messages to and from the VANguard service that include references to algorithms and/or ciphers that are not aligned with the Australian Government Information Security Manual (ISM). These are included for illustration only and should not be taken as a statement of VANguard's technical capability. Please contact VANguard for the most up-to-date details of supported algorithms and ciphers.

http://vanguard.ebusiness.gov.au/2008/06/identity/claims/credentialtype	Credential Type (ABR_User or ABR_Device)	Mandatory	Mandatory
http://vanguard.ebusiness.gov.au/2008/06/identity/claims/samlsubjectid	SAML subject ID (composite global unique ID)	Mandatory	Mandatory
http://vanguard.ebusiness.gov.au/2008/06/identity/claims/fingerprint	SHA1 hash of the business certificate.	Mandatory	Mandatory
http://vanguard.ebusiness.gov.au/2008/06/identity/claims/sbr_personid	Sender user ID (unique serial within one business)	Mandatory	Not applicable
http://vanguard.ebusiness.gov.au/2008/06/identity/claims/givennames	User given names	Mandatory	Not applicable
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	User family name	Mandatory	Not applicable
http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	User e-mail address	Mandatory	Not applicable
http://vanguard.ebusiness.gov.au/2008/06/identity/claims/credentialadministrator	Boolean indicator if user is administrator	Mandatory	Not applicable
http://vanguard.ebusiness.gov.au/2008/06/identity/claims/previoussubject	ID of any previous credential subject ID (eg an ATO certificate)	Optional	Not applicable
http://vanguard.ebusiness.gov.au/2008/06/identity/claims/stalecrminutes	Client credential was checked against a Certificate Revocation List that was overdue for replacement by this many minutes.	Optional	Optional
http://vanguard.ebusiness.gov.au/2008/06/identity/claims/subjectdn	User X.509 distinguished name	Optional	Optional
http://vanguard.ebusiness.gov.au/2008/06/identity/claims/issuerdn	Issuer (ABR) X.509 distinguished name	Optional	Optional
http://vanguard.ebusiness.gov.au/2008/06/identity/claims/notafterdate	Certificate validity expiry date time	Optional	Optional
http://vanguard.ebusiness.gov.au/2008/06/identity/claims/certificateserialnumber	Certificate serial number	Optional	Optional

Table 4: STS Claims

Note that the client software does not need to provide values for these claims. The Token.Request message simply lists the claim URIs shown above and the STS will return the values as a set of assertions within an encrypted token.

### 5.1.1. STS Request HTTPS Connection Endpoint

The client software must establish a HTTPS connection to the appropriate endpoint using the standard port 443.

Environment	STS Endpoint
Production	https://authentication.business.gov.au/R3.0/vanguard/S007v1.2/service.svc
Test	https://thirdparty.authentication.business.gov.au/R3.0/vanguard/S007v1.2/service.svc

### 5.1.2. Creating the STS Request

This section defines the structure of the Token.Request message. The XML snippet below is an example Token.Request sent to the STS.

Notes:

- the base64 data representing the business certificate has been removed for readability.
- Ciphers and hashing algorithms shown in the examples are for illustration only. Please refer to VANguard for details of their supported ciphers and hashing algorithms.

```
<?xml version='1.0' encoding='utf-8'?>
<soapenv:Envelope xmlns:soapenv="http://www.w3.org/2003/05/soap-envelope">
  <soapenv:Header xmlns:wsa="http://www.w3.org/2005/08/addressing">
    <wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
soapenv:mustUnderstand="true">
      <wsu:Timestamp xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
wsu:Id="Timestamp-19714461">
        <wsu:Created>2009-10-12T04:02:23.890Z</wsu:Created>
        <wsu:Expires>2009-10-12T04:07:23.890Z</wsu:Expires>
      </wsu:Timestamp>
      <wsse:BinarySecurityToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
utility-1.0.xsd" EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
1.0#Base64Binary" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
wsu:Id="CertId-11658721"><!-- Binary data removed --></wsse:BinarySecurityToken>
        <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#" Id="Signature-620055">
          <ds:SignedInfo>
            <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
            <ds:Reference URI="#Timestamp-19714461">
              <ds:Transforms>
                <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
              </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <ds:DigestValue>RUqAWdHk+v4Xkx+9Sw0HLVKijpE=</ds:DigestValue>
          </ds:Reference>
          <ds:Reference URI="#id-3125250">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <ds:DigestValue>8M8SCNohyZHQEHzru8hIzZludu8=</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
      </ds:SignatureValue>
    </wsse:Security>
  </soapenv:Header>
  <!-- Body content -->
</soapenv:Envelope>
```

```

ZHSrVAqDjIzZLdFaz9gEnXFuvpf/q3rQ9XWI3Bu9ZtVkuubhvXXnQo1zeIyiY84uiS/J5ILMpqTN
dQv5JRWZk6Y6XzrEQfofp5VSmDeyNgVXJtm2FFePnYq7OTZerA3c7jhQO9xc0MpNLfs8NX7zCnl
UNbhgeKR6LFG8EW1hU=
  </ds:SignatureValue>
  <ds:KeyInfo Id="KeyId-15834478">
    <wsse:SecurityTokenReference xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" wsu:Id="STRId-7789321">
      <wsse:Reference URI="#CertId-11658721" ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-x509-token-profile-1.0#X509v3" />
    </wsse:SecurityTokenReference>
  </ds:KeyInfo>
</ds:Signature>
</wsse:Security>
<wsa:To xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" wsu:Id="id-
3125250">https://thirdparty.authentication.business.gov.au/R3.0/vanguard/S007v1.1/service.svc</wsa:To>
<wsa:MessageID>urn:uuid:CC8BEAE32759FDD5821255320143481</wsa:MessageID>
<wsa:Action>http://docs.oasis-open.org/ws-sx/ws-trust/200512/RST/Issue</wsa:Action>
</soapenv:Header>
<soapenv:Body>
  <wst:RequestSecurityToken xmlns:wst="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
    <RequestType xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512">http://docs.oasis-open.org/ws-sx/ws-
trust/200512/Issue</RequestType>
    <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
      <EndpointReference xmlns="http://www.w3.org/2005/08/addressing">
        <Address>https://m2m.uat.asic.gov.au</Address>
      </EndpointReference>
    </wsp:AppliesTo>
    <TokenType xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512">http://docs.oasis-open.org/wss/oasis-wss-
saml-token-profile-1.1#SAMLV2.0</TokenType>
    <Claims xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
xmlns:i="http://schemas.xmlsoap.org/ws/2005/05/identity" Dialect="http://schemas.xmlsoap.org/ws/2005/05/identity">
      <i:ClaimType Optional="false" Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/abn" />
      <i:ClaimType Optional="false" Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/commonname" />
      <i:ClaimType Optional="false" Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/credentialtype" />
      <i:ClaimType Optional="false" Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/samlsubjectid" />
      <i:ClaimType Optional="false" Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/fingerprint" />
      <i:ClaimType Optional="true" Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/sbr_personid" />
      <i:ClaimType Optional="true" Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/givennames" />
      <i:ClaimType Optional="true" Uri="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname" />
      <i:ClaimType Optional="true" Uri="http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress" />
      <i:ClaimType Optional="true"
Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/credentialadministrator" />
      <i:ClaimType Optional="true" Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/stalecrlminutes" />
      <i:ClaimType Optional="true" Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/subjectdn" />
      <i:ClaimType Optional="true" Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/issuerdn" />
      <i:ClaimType Optional="true" Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/notafterdate" />
      <i:ClaimType Optional="true"
Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/certificateserialnumber" />
      <i:ClaimType Optional="true" Uri="http://vanguard.ebusiness.gov.au/2008/06/identity/claims/previoussubject" />
    </Claims>
    <Lifetime xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
      <wsu:Created xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">2009-10-12T04:02:22.984Z</wsu:Created>
      <wsu:Expires xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">2009-10-12T04:32:22.984Z</wsu:Expires>
    </Lifetime>
    <KeyType xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512">http://docs.oasis-open.org/ws-sx/ws-
trust/200512/SymmetricKey</KeyType>
    <KeySize xmlns="http://docs.oasis-open.org/ws-sx/ws-trust/200512">512</KeySize>
  </wst:RequestSecurityToken>
</soapenv:Body>
</soapenv:Envelope>

```

Figure 8: Token.Request Sample Message

The Token.Request message includes the following elements:

- The `soapenv:Header` element contains the WS-Addressing elements that define the location end-point of the STS, a unique message ID, and the requested action (to Issue a token).
- The `<wst:RequestSecurityToken>` element in the SOAP Body defines the specifics of the request to the STS and includes the following key elements:



- The <wsp:AppliesTo> element defines the location of the service that requires the SAML token as per table below.

Environment	Applies-to Address
Production	https://m2m.asic.gov.au
Test	https://m2m.uat.asic.gov.au

- The <TokenType> element defines the required token as a SAML 2.0 token.
- The <Claims> element and all the related <i:ClaimType> elements specify exactly which identity assertions the STS is to include in the Token.Response message (as encrypted assertions).
- The <Lifetime> element defines the required lifetime of the session key to be returned by the STS. Note that the STS may enforce a maximum allowed validity period and so it is possible that the lifetime actually provided may be less than the lifetime requested. The minimum lifetime for a session key will be 30 minutes.
- The <KeyType> element and <KeySize> element specify that the returned session key should be a 512 bit symmetric key.

### 5.1.3. Processing the STS Response

This section provides guidelines on how client software should process the security token and session key that is returned from the STS in the Token.Response message.

The XML snippet below is an example response from the STS.

Notes:

- the base64 data representing the encrypted token has been removed for readability.
- Ciphers and hashing algorithms shown in the examples are for illustration only. Please refer to VANGUARD for details of their supported ciphers and hashing algorithms.

```
<?xml version='1.0' encoding='utf-8'?>
<s:Envelope xmlns:s="http://www.w3.org/2003/05/soap-envelope" xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd" xmlns:a="http://www.w3.org/2005/08/addressing">
  <s:Header>
    <a:Action s:mustUnderstand="1">http://docs.oasis-open.org/ws-sx/ws-trust/200512/RSTRC/IssueFinal</a:Action>
    <a:RelatesTo>urn:uuid:CC8BEAE32759FDD5821255320143481</a:RelatesTo>
    <ActivityId xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics" CorrelationId="6da8a0dc-89cf-4210-9d6b-36b16ffb3788">00000000-0000-0000-0000-000000000000</ActivityId>
    <o:Security xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" s:mustUnderstand="1">
      <u:Timestamp u:Id="_0">
        <u:Created>2009-10-12T04:02:26.742Z</u:Created>
        <u:Expires>2009-10-12T04:07:26.742Z</u:Expires>
      </u:Timestamp>
    </o:Security>
  </s:Header>
  <s:Body>
    <trust:RequestSecurityTokenResponseCollection xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512">
      <trust:RequestSecurityTokenResponse>
        <trust:KeySize>512</trust:KeySize>
        <trust:Lifetime>
          <wsu:Created xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">2009-10-12T04:02:26.454Z</wsu:Created>
          <wsu:Expires xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">2009-10-12T04:32:22.984Z</wsu:Expires>
        </trust:Lifetime>
      </trust:RequestSecurityTokenResponse>
    </trust:RequestSecurityTokenResponseCollection>
  </s:Body>
</s:Envelope>
```



```

<wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
  <a:EndpointReference>
    <a:Address>https://m2m.uat.asic.gov.au</a:Address>
  </a:EndpointReference>
</wsp:AppliesTo>
<trust:RequestedSecurityToken>
  <EncryptedAssertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion">
    <xenc:EncryptedData xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
      <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc" />
      <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
        <e:EncryptedKey xmlns:e="http://www.w3.org/2001/04/xmlenc#">
          <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p">
            <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
          </e:EncryptionMethod>
          <KeyInfo>
            <o:SecurityTokenReference xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
              <X509Data>
                <X509IssuerSerial>
                  <X509IssuerName>CN=Australian Government Notary
Services OCA, OU=For Development purposes ONLY, OU=Australian Authentication and Notary Services, O=Australian
Government, C=AU</X509IssuerName>
                <X509SerialNumber>116425329959729741023280816821386492610</X509SerialNumber>
              </X509IssuerSerial>
            </X509Data>
          </o:SecurityTokenReference>
        </KeyInfo>
      <e:CipherData>
        <e:CipherValue>G10F8BxPQNcshpWwiPgvoofH74lSiNpL1h9bP4pZPHwyxrlO+xirH5XAMqi+BkTCBbFojAEJaYIvu9NIqF
Z8THUZVlhbhBge6miNrsx+kRz70+QJKv6F9WmCmH+RqWNi1T1MjEP9xlcVLCzDzvFBPEJlvK13KLV5Hoimxp/8F8=</e:Cip
herValue>
      </e:CipherData>
    </e:EncryptedKey>
  </KeyInfo>
<xenc:CipherData>
  <xenc:CipherValue><!-- Binary data removed --></xenc:CipherValue>
</xenc:CipherData>
</xenc:EncryptedData>
</EncryptedAssertion>
</trust:RequestedSecurityToken>
<trust:RequestedProofToken>
  <trust:BinarySecret>DolCv6k0OrHRKqMZA5AgH28SC7ntQN1EgOXybaYq9GYh3ppK6gfpuHRR4NplJo2sEcnQ6+djWRs
8orObzXDclQ==</trust:BinarySecret>
</trust:RequestedProofToken>
<trust:RequestedAttachedReference>
  <SecurityTokenReference xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:b="http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd"
b:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0">
    <KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLID">_d5511ae3-5ab6-474a-b58f-5752b847ab15</KeyIdentifier>
  </SecurityTokenReference>
</trust:RequestedAttachedReference>
<trust:RequestedUnattachedReference>
  <SecurityTokenReference xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-secext-1.0.xsd" xmlns:b="http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd"
b:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0">
    <KeyIdentifier ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLID">_d5511ae3-5ab6-474a-b58f-5752b847ab15</KeyIdentifier>
  </SecurityTokenReference>
</trust:RequestedUnattachedReference>
<trust:TokenType>http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
1.1#SAMLV2.0</trust:TokenType>
<trust:RequestType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/Issue</trust:RequestType>
<trust:KeyType>http://docs.oasis-open.org/ws-sx/ws-trust/200512/SymmetricKey</trust:KeyType>
</trust:RequestSecurityTokenResponse>
</trust:RequestSecurityTokenResponseCollection>
</s:Body>
</s:Envelope>

```

Figure 9: Token.Response

The token response includes the following elements:

- a. The `<trust:RequestSecurityTokenResponseCollection>` is the envelope element that contains all the token data. It contains one `<trust:RequestSecurityTokenResponse>`.
- b. The `<trust:KeySize>` element defines the key size of the symmetric session key. The session key is used to sign envelopes sent to ASIC. This is for information only.
- c. The `<trust:Lifetime>` element defines the validity period for the symmetric session key. The client should request a new session key before expiry in order to continue any interactions with ASIC.
- d. The `<wsp:AppliesTo>` element is the identifier for the service end point (in this case ASIC) that is the “relying party” in the WS-Trust interaction. This is for information only.
- e. The `<trust:RequestedSecurityToken>` element contains the `<EncryptedAssertion>` structure. The client should insert the entire `<EncryptedAssertion>` structure into the SOAP header of the Service.Request envelope as described in 5.4. This element contains all the identity claims (encrypted) together with the necessary key reference information that allows ASIC to decrypt the claims.
- f. The `<trust:RequestedProofToken>` element contains the symmetric key that is used by the client to sign interactions with ASIC until it expires and must be renewed.
- g. The `<trust:RequestedAttachedReference>` element contains the unique identifier for the symmetric session key (the `<trust:RequestedProofToken>` ). This is not currently used.
- h. The `<trust:TokenType>` (SAML2), `<trust:RequestType>` (Issue), and `<trust:KeyType>` (Symmetric) elements contain informational data about the nature of the Token.Response. No specific action is required from client software.

## 5.2. USERNAME & PASSWORD AUTHENTICATION

In order to use this authentication method, users must obtain a Username and Password in order to use M2M Web Services exposed by ASIC. Details of the registration process are available on by contacting [webservices@asic.gov.au](mailto:webservices@asic.gov.au).

ASIC requires that all requests use HTTP Basic Authentication. Basic Authentication was originally defined by RFC 1945[2] (Hypertext Transfer Protocol – HTTP/1.0) although further information regarding security issues may be found in RFC 2616 (Hypertext Transfer Protocol – HTTP/1.1) and RFC 2617 (HTTP Authentication: Basic and Digest Access Authentication).

ASIC expects a Basic Authentication header to be included pre-emptively in all requests. If requests are sent with no Basic Authentication header, ASIC does not reply with a HTTP/1.1 401 Authorization Required error. Instead we reply with a valid HTTP SOAP response containing ASIC error 00005 Authentication error.

```
<business.document.header.types:messageEvents>
  <business.document.header.types:messageEvent>
    <business.document.header.types:errorCode>00006</business.document.header.types:errorCode>
    <business.document.header.types:serverityCode>Error</business.document.header.types:serverityCode>
    <business.document.header.types:description>Authentication
error</business.document.header.types:description>
    <business.document.header.types:details>Rejected by policy</business.document.header.types:details>
  </business.document.header.types:messageEvent>
```

```
</business.document.header.types:messageEvents>
```

**Figure 10: Sample XML - Reply authentication error event**

### 5.3. SECURE MESSAGING

ASIC Web Services Security has the following goals:

- To ensure confidentiality of business data. This is achieved through transport layer security (SSL) using the Port 443.
- To ensure tamper proofing and non-repudiation of origin for messages. This is achieved through a digital signature of the business payload
- To support identification and authentication of business users or other governmental agencies. This is supported through the STS call and SAML assertions within the encrypted token.
- To support non-repudiation of receipt by ASIC. This is achieved through digital signature of the response message.
- Data encryption is compliant with the Australian Government Information Security Manual (ISM). Specifically:
  - we use 2048-bit RSA keys (ie modulus of 2048) for encryption
  - SHA-1 hashing is disabled
  - AES Electronic Cookbook mode is disabled

The message security implementation employs standards defined in the WS-I Basic Security Profile v1.1.

### 5.4. SIGNATURE STRUCTURES

The sample envelope below provides an example of the ASIC web services security header. Note that carriage returns have been inserted and Base64 strings have been truncated for readability.

The SOAP header contains one <wsse:Security> structure. This structure contains four main sub components:

- A <saml2:EncryptedAssertion> that carries identity information for the agency. This information is provided by the STS and forwarded unchanged to the agency.
- A <wsse:BinarySecurityToken> that carries the business certificate. This is used by the agency to validate the document signature.
- A <ds:Signature id="signed\_request\_id"> element that carries the signature of the SOAP Body and the business certificate. Signed with the business certificate. This signature is for non-repudiation of origin.
- A <ds:Signature id="soapbody\_id"> element that carries the enveloping signature of the SOAP Body, the SAML Token, and the business document signature. This signature links the business document to the SAML identity token.

All these structures are described in more detail in subsequent sections.

Please note that the examples below include SHA-1 as a hashing algorithm. In practice this algorithm is deprecated and will be phased out in favour of other more secure algorithms. Updated details will be provided in a subsequent release of the WIG.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soap:Header>
    <wsse:Security
      soap:MustUnderstand="1"
      xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <saml2:EncryptedAssertion
        wsu:Id="assertion_id"
        xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">
        <xenc:EncryptedData
          xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
          xmlns:a="http://www.w3.org/2005/08/addressing"
          xmlns:s="http://www.w3.org/2003/05/soap-envelope"
          xmlns:trust="http://docs.oasis-open.org/ws-sx/ws-trust/200512"
          xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
          xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
          <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc"/>
          <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
            <e:EncryptedKey xmlns:e="http://www.w3.org/2001/04/xmlenc#">
              <e:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p">
                <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
              </e:EncryptionMethod>
              <KeyInfo>
                <o:SecurityTokenReference
                  xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
                  <X509Data>
                    <X509IssuerSerial>
                      <X509IssuerName>
                        CN=Australian Government Notary Services OCA,
                        OU=For Development purposes ONLY,
                        OU=Australian Authentication and Notary Services,
                        O=Australian Government, C=AU
                      </X509IssuerName>
                      <X509SerialNumber>116425329959729741023280816821386492610
                      </X509SerialNumber>
                    </X509IssuerSerial>
                  </X509Data>
                </o:SecurityTokenReference>
              </KeyInfo>
            <e:CipherData>
              <e:CipherValue> TdTH/IV7gccbRW2PWou/Cch7P8g8aOqjy35W9nyreTwE=</e:CipherValue>
            </e:CipherData>
          </e:EncryptedKey>
        </KeyInfo>
      <xenc:CipherData>
        <xenc:CipherValue> / +euKyWoJmES +ghWa/hnSkMPvHQtn6B0sSSTAVJu5c=</xenc:CipherValue>
      </xenc:CipherData>
    </xenc:EncryptedData>
  </saml2:EncryptedAssertion>
  <wsse:BinarySecurityToken
    EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary"
    ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"
    wsu:Id="BinarySecurityToken-78bd05bf-80de-4bbf-9084-8fb3ee78f423">
    MIIHDCCAwwSgAwIBAgICCKMwDQYJKoZIhvcNAQEFBQAwwYUxCzAJB
  </wsse:BinarySecurityToken>
  <ds:Signature Id="signed_request_id" xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <ds:SignedInfo Id="signedInfo-eac65786-48f7-499b-a688-b1328de4ef9a">
      <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
      <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
      <ds:Reference URI="#request_id">
        <ds:Transforms>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>kvKBax6HB1cVffej7WggalQ8DM4=</ds:DigestValue>
        </ds:Reference>
      </ds:Signature>
    </ds:SignedInfo>
  </ds:Signature>
</soap:Header>
</soap:Envelope>
```

```

        <ds:Reference URI="#BinarySecurityToken-78bd05bf-80de-4bbf-9084-8fb3ee78f423">
            <ds:Transforms>
                <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>R3KAVHwhs7kRpwRG6WrPnOEomCo=</ds:DigestValue>
        </ds:SignedInfo>
        <ds:SignatureValue Id="signatureValue-46341d8d-e8b4-4e83-85e8-47dc6539fe77">
            iawKJTf1/elSjy3aX0mhJ+AORQp2mgkeogwD5tsT/TPlcNGgmJOCw7cx6YmDDJDBzwMVbrVkJCyY
            mDwJW1a/T7hxW+/WbaA8t0rLE0lJPuxUDzCzUJdZx/KJy5KGt8GVZ8n6k0BSB8hCeF444Ircggxv
            05NaSZ8DbopHUuQ6hNk=</ds:SignatureValue>
        <ds:KeyInfo Id="KeyInfo-f4a3af05-89be-444d-8d91-36c66f23f222">
            <wsse:SecurityTokenReference>
                <wsse:Reference
                    URI="#BinarySecurityToken-78bd05bf-80de-4bbf-9084-8fb3ee78f423"
                    ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3"/>
            </wsse:SecurityTokenReference>
        </ds:KeyInfo>
    </ds:Signature>
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
            <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
            <ds:Reference URI="#signed_request_id">
                <ds:Transforms>
                    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <ds:DigestValue>v0dxw6ZPMBV/Pjs6dXUrJb5KMo=</ds:DigestValue>
            </ds:Reference>
            <ds:Reference URI="#soapbody_id">
                <ds:Transforms>
                    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <ds:DigestValue>9YrzsQ8dVHJltSzvyuOLK6pUxWY=</ds:DigestValue>
            </ds:Reference>
            <ds:Reference URI="#assertion_id">
                <ds:Transforms>
                    <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
                </ds:Transforms>
                <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
                <ds:DigestValue>/MmViMAR3/2gmJcXh6JQivjn86o=</ds:DigestValue>
            </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>aNuW8eO5gWKAMDDP+45iP4XkMNY=</ds:SignatureValue>
    </ds:Signature>
    </wsse:Security>
</soap:Header>
<soap:Body wsu:Id="soapbody_id">
    .... insert content here
</soap:Body>
</soap:Envelope>

```

Figure 11: Sample Security Header

### 5.4.1. Identity Token <saml2:EncryptedAssertion>

This structure contains the list of assertions that provide identity information to the ASIC. The actual assertions are represented as an encrypted string contained in the <xenc:CipherValue> element near the end of the structure. The remainder of the elements provide the agency with the necessary data to decrypt the assertions:

- The assertions are encrypted with a symmetric key that must be passed to the agency. The symmetric key is itself encrypted using the public key of the agency. The <e:CipherValue> element just above the encrypted assertions contains the encrypted symmetric key.

- The `<o:SecurityTokenReference>` element provides the id reference of the public key used to encrypt the symmetric key. The agency must use the corresponding private key to decrypt the symmetric key. The agency can then use the decrypted symmetric key to decrypt the identity assertions.
- The encryption algorithm for the assertions is AES-256 (symmetric key encryption). The encryption algorithm for the symmetric key is RSA (asymmetric key encryption).

The structure is part of the Token.Response from the STS and can be inserted into the Service.Request security header without change.

#### 5.4.2. Business Certificate `<wsse:BinarySecurityToken>`

This structure contains the digital certificate of the business. The certificate is encoded as a base64 string and is identified using the attribute `wsu:id="BinarySecurityToken-78bd05bf-80de-4bbf-9084-8fb3ee78f423"`. The agency will use the public key contained in this certificate to validate the business document signature.

#### 5.4.3. Document Signature `<ds:Signature id="signed_request_id">`

This structure contains the digital signature of the business document. There are three main sub-structures:

- The `<ds:SignedInfo>` element identified the parts of the SOAP envelope that are signed. These parts are referenced using the `ds:Reference URI` elements that locate elements within the envelope identified by a `wsu:id`.
- The `<ds:SignatureValue>` element contains the actual digital signature.
- The `<ds:KeyInfo>` element contains the reference to the certificate used to create the signature. In this case the certificate is identified by reference to the business certificate described in the previous section using the `<wsse:Reference URI="#BinarySecurityToken-78bd05bf-80de-4bbf-9084-8fb3ee78f423">` element.

#### 5.4.4. Envelope Signature `<ds:Signature>`

This structure contains the envelope signature created using the symmetric key provided by the STS. There are two sub-structures:

- The `<ds:SignedInfo>` element identified the parts of the SOAP envelope that are signed. These parts are referenced using the `ds:Reference URI` elements that locate elements within the envelope identified by a `wsu:id`. In the case of the envelope signature, the signed parts are the business document signature (`<ds:Reference URI="#signed_request_id">`), the SOAP Body (`<ds:Reference URI="#soapbody_id">`), and the encrypted SAML Token (`<ds:Reference URI="#assertion_id">`).
- The `<ds:SignatureValue>` element contains the actual digital signature.

There is no `<ds:KeyInfo>` element because this signature is created using the STS session key. The encrypted session key is provided for the agency within the encrypted assertions.



## 6. TESTING

### 6.1. OVERVIEW

ASIC web services offers several options to assist the software developers in the testing of their services, allowing a software developer to “step up” from basic tests to more sophisticated tests. The options are:

1. Network connectivity testing
2. Message connectivity testing
3. Message end-to-end interaction testing.

Further explanation of these options is provided in subsequent sections.

#### 6.1.1. Service End Points

The MIGs will cover in details the WSDL URLs at which each of the service is available.

There are two environments to which software developers have access, one for testing and one for production. Message connectivity tests may also be performed in the production environment as part of any diagnostic functions within a software package.

Test and production end points for the Security Token Service are also provided at the locations documented in the section 5.1.1.

### 6.2. NETWORK CONNECTIVITY TESTING

BN conforms to the industry convention of returning the WSDL for a given service if an HTTP GET request is performed on the service URL with the string “?wsdl” appended to it. This applies to any of the end points documented in the Message Implementation Guides.

For example, requesting the URL “http://registry.asic.gov.au/services/XXXservice?wsdl” in a web browser will confirm that network connectivity is present between the network on which the browser is operating and the production ‘XXX’ service.

This environment does not require any authentication or message level security to be added as part of the request, and will generate a SOAP fault if such information is provided.

### 6.3. MESSAGE CONNECTIVITY TESTING

#### 6.3.1. Overview

Having generated syntactically valid SOAP Body structures, the next phase in development is to add the security solution to messages, and confirm that it is working correctly. The code already developed and tested against the service, can now be upgraded, and the end points against which it is invoked modified as detailed in the above table.

To assist in the testing of the security implementation, ASIC offer a simple message connectivity test, called “ping”, which is designed as an end-to-end connectivity test from business software, to ASIC and back. ‘ping’ is not a separate web service; it is implemented as a particular message type that is

supported by each of the ASIC BN web services. ‘ping’ is implemented in ASIC BN test and production systems.

- In the ASIC web services test environment, software developers may use the “ping” to test that they have correctly implemented all BN messaging and security protocols
- In the ASIC web services production environment, business users can use “ping” as a diagnostic tool in the event of difficulties with business functions. Software developers are encouraged to include this connectivity test as a diagnostic capability within their product release.

### 6.3.2. ping

The "ping" re-uses the existing services. Therefore it must also be authenticated like any other interaction.

- For AUSkey authentication, the client must call the Security Token Service (STS) for a SAML token and construct a security header in accordance with the security specification in this document (section 5.1)
- For username/password authentication the client must pass a valid username and password combination in accordance with the security specification this document (see section 5.2).

A successful ping will result in the return of a “ping” that is a copy of the ping with the addition of timestamps

Any transport level condition will result in the return of a SOAP fault with error codes as defined in the error handling section of this document.

ping can be sent using the WSDLs for the ping service listed in the table below.

ENVIRONMENT	END POINT
Test	https://m2m.uat.asic.gov.au/gateway/ExternalPingPort?WSDL
Production	https://m2m.asic.gov.au/gateway/ExternalPingPort?WSDL

**Table 5: WSDLs for Ping Service**

The table below provides specific details on how to populate a request message for the ping service.

SOAP ELEMENT	ELEMENT VALUE
messageType	“ping”

**Table 6: Specific Values For ping**



### 6.3.3. ping reply

The pingReply envelope is returned by the target agency in response to a successful ping. The pingReply is essentially an echo of the ping with additional timestamp information in the Business Document Header. The message pingReply follows the standard message response structure defined previously. The table below provides details on how the fields specific to pingReply will be populated by the ASIC.

SOAP ELEMENT	ELEMENT VALUE
messageType.	"ping"
messageTimestamp (generated by requestor)	Date/time as provided on ping request
messageTimestamp (generated by ASIC)	Date/time response envelope was created

**Table 7: Specific values for ping reply**

## 6.4. MESSAGE END-TO-END INTERACTION TESTING

Having determined that SOAP messages can be successfully generated, secured and sent to ASIC, and that the resulting response can be interpreted, full testing of the desired services can commence.

ASIC offers a range of artefacts to assist the end-to-end testing such as: test credentials (i.e. test AUSkey certificates or username & password), test cases and test data.

## 7. ASIC WEB SERVICES REGISTRATION

Clients are expected to register in order to have access to the documentation, schema files and WSDL. Upon registration the users will be provided with a unique identifier (i.e. senderId) that must be used each time they initiate a request using the ASIC web services.

Please contact ASIC at [webservices@asic.gov.au](mailto:webservices@asic.gov.au) for information on how to register to use the ASIC web services.

# INDEX

## A

ABN ..... 8, 19, 28  
ABR ..... 8, 11, 27, 29  
AUSkey .....8, 9, 11, 13, 27, 40  
Australian Business Number .....See ABN  
Australian Business Register..... See ABR  
Australian Government Information Security Manual See  
ISM  
Authentication.....8, 9, 13, 27, 34, 39, 40

## C

copyright ..... 2

## I

ISM ..... 8, 28, 35

## M

Message Implementation Guide .....See MIG  
MIG..... 8, 13, 18, 19, 20

## P

Password .....10, 13, 27, 34, 40, 41

## S

Secure Socket Layer..... See SSL  
Secure Token Service..... See STS  
SOAP .....6, 8, 11, 13, 14, 15, 20, 22, 23, 24, 25, 26, 27, 28,  
31, 34, 35, 38, 39, 40, 41  
SSL ..... 12, 28, 35  
STS...8, 9, 10, 11, 12, 27, 28, 29, 30, 31, 32, 35, 38, 39, 40

## U

Username .....10, 13, 27, 34, 40, 41

## V

VANguard .....8, 9, 10, 11, 24, 28, 30, 32

## W

Web Services Description Language..... See WSDL  
WSDL ..... 7, 9, 10, 11, 14, 24, 39, 40, 41