ANZLIC
the Spatial Information Council

National Address Management Framework

Web Services Developer's Guide

Document Control

Author: Bronwyn Cook, Brett Walker, Steven Smith

Geometry Pty Ltd

Document No: 4 **Date of Issue:** 26 July 2010

Version: 1.0

Document Release Information and Distribution

This is a controlled document. The Quality Document Register is maintained within CTG to manage the distribution of controlled documents and the issue of any revisions. Printed versions are uncontrolled.

Date	Version	Released To
16 June 2010	0.1	PSMA Australia
13 July 2010	0.2	PSMA Australia (added addressIdentifier to listAddress)
19 July 2010	0.3	GovDex – minor change to listAddress example
26 July 2010	1.0	First official release. Added duplicateHandling attribute to getNotifications

Table of Contents

1 Introduction

This documentation reflects version 1.0 of the NAMF Web Service.

1.1 General Notes

In the following document the following are implied throughout:

- All names and values of webservice elements are case-sensitive.
- The reference dataset is the latest GNAF and the latest PAF.

1.2 Requests and Responses

The philosophy of the NAMF Web Service is to submit a request and receive a response via the one execute operation defined in the WSDL[1]. This WSDL is designed to support future needs and demands by providing a simple infrastructure upon which a variety of different types of requests and responses can be channelled through.

```
<xsd:element name="execute" type="execute" />

<xsd:element name="executeResponse" type="executeResponse" />

<xsd:complexType name="execute">
  <xsd:sequence>
    <xsd:element minOccurs="0" ref="namf:requests"/>
  </xsd:sequence>
</xsd:complexType>

<xsd:complexType name="executeResponse">
  <xsd:sequence>
    <xsd:element minOccurs="0" ref="namf:responses"/>
  </xsd:sequence>
</xsd:complexType>
```

XML Schema extract from the NAMF WSDL

1.3 Requests

The `requests` element provides a mechanism to submit multiple requests, if so desired. Alongside the list of requests (`request` elements) are also the credentials, a list of global features, the version and the identifier.

How an individual request is specified is discussed in later sections of the document. The list of requests can be empty as this is needed to support asynchronous processing.

The identifier (`id` attribute) is user-defined and is not used by any NAMF Web Service function. It only provides as means for users to identify their requests submitted for their own purposes.

The version (`version` attribute) is the version of the NAMF Web Service that processed the individual requests. If the version string cannot be interpreted the default version being the latest

[1] This is the most common 'synchronous' request. The webservice may also support 'asynchronous' processing, discussed later on in this document.

version will be used. If this happens the user will be notified using the notes functionality provided in the `responses` element (no error will be thrown). See the **Versions** section for more information.

The list of global features (`features` element) provides the means to specify features that will apply to all requests. This is particularly useful when multiple requests of a similar nature are submitted. Any feature specified at the request level overrides the feature at the global level, if it is single occurrence feature; otherwise it extends the feature at the global level for multiple occurrence features. See the **Features and Attributes** section in this document for more information.

The credentials (`authentication` element) are used for authentication in order to allow access to the NAMF Web Service functions. See the **Security and Authentication** section in this document for more information.

```
<xs:element name="requests">
    <xs:annotation>
      <xs:documentation>A Requests Object is the top-level object set to
the NAMF Web Services.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="authentication">
          <xs:complexType>
            <xs:sequence>
                    <xs:element name="username" type="xs:string" />
                    <xs:element name="password" type="xs:string" />
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
              <xs:element ref="features" />
              <xs:element minOccurs="0" maxOccurs="unbounded"
name="request">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element minOccurs="0" ref="features" />
                    <xs:element minOccurs="0" ref="address" />
                    <xs:element minOccurs="0" ref="attributes" />
                  </xs:sequence>
                  <xs:attribute name="id" type="xs:string" />
                  <xs:attribute name="name" type="xs:string" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute name="id" type="xs:string" />
            <xs:attribute name="version" type="xs:string" />
          </xs:complexType>
</xs:element>
```

XML Schema extract for the requests element from the NAMF XSD.

1.4 Responses

The responses element provides a mechanism to return multiple responses. Alongside the list of responses (response elements) are also a list of global notes, a result status, a list of errors and an identifier.

How an individual response is specified is discussed in later sections of the document.

The identifier (id attribute) is the same identifier provided in the requests .

The list of errors (error elements) contains all the errors encountered while processing the requests as a whole. Some examples of possible types of errors are an authentication error, a severe database error or a severe server error. An error encountered while processing an individual request will not be found in this error list. This list of errors can be empty indicating that no errors occurred.

The result status (status attribute within the result element) indicates the status of processing the list of requests (an individual request could have an error). A result status having a status of OK and completed set to true will indicate a successful processing of all the requests. A result status having a status of ERROR and completed set to false will indicate an unsuccessful processing of all the requests.

The list of notes (notes element) contains all the possible messages encountered while processing the requests as a whole. An example of a possible type of note is a version message indicating a wrong version specification and that it is using a default version instead.

```
<xs:element name="responses">
    <xs:annotation>
      <xs:documentation>A Responses Object is the top-level object returned by the NAMF Web
Services.</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" ref="notes" />
        <xs:element name="result">
          <xs:complexType>
            <xs:sequence>
              <xs:element minOccurs="0" ref="attributes" />
            </xs:sequence>
            <xs:attribute name="status" type="statusValue" use="required" />
            <xs:attribute name="completed" type="xs:boolean" use="required" />
            <xs:attribute name="hasErrorsInResponseElements" type="xs:boolean" use="required"
/>
          </xs:complexType>
        </xs:element>
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="error" />
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="response" />
      </xs:sequence>
      <xs:attribute name="id" type="xs:string">
        <xs:annotation>
          <xs:documentation>The id from the origin</xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:complexType>
</xs:element>
```

XML Schema extract for the responses element from the NAMF XSD.

1.5 Security and Authentication

Users must be registered to use the NAMF Web Service. The user's credentials (username & password) can either be sent as plaintext (in which case a secure SSL connection is strongly recommended) or as plaintext username and MD5 encrypted password.

1.6 Versions

The NAMF Web Service is likely to evolve by including new functionality or altering existing functionality. In order to support backwards compatibility each new set of functional specifications for the NAMF Web Service will be enabled in a new version. Thus users interacting with the NAMF Web Service at version X will not have any changed or reduced functionality by the introduction of a new version Y. The only exception to this is if a significant security-related issue is fixed in version Y the functionality of version X may change or version X may be completely disabled.

A version is specified by a numeric string of the form x.y where x is an integer representing the major version number and y is an integer representing the minor version number. The role of the major version number is to control the set of NAMF Web Service functions that are available. A new major version will be required for one of the following reasons:

- The inclusion of a new Web Service function;
- The deprecation of an existing Web Service function or
- The removal of a deprecated Web Service function.

The role of the minor version is to control the functionality of a set of NAMF Web Service functions. A new minor version will be required for one of the following reasons:

- Change to a Web Service function's set of attributes;
- Change to a Web Service function's set of features or
- Significant change to a Web Service function's behaviour.

1.7 Features and Attributes

Features and attributes are used throughout the NAMF Web Service to supply additional arguments or data for the functions. What is a feature and what is an attribute is difficult to define. As a general rule a feature controls the processing of data; whereas an attribute is data to be processed, its value has been stored (used for retrieval) or is to be stored.

The usage of a feature or attribute in one function does not imply it usage in another function. In a similar vein, a feature name X in one function does not imply a relationship to an attribute name X in another function just because of its name alone. Thus the feature and attribute set for a function should be considered unique and distinct unless indicated otherwise (There are a significant number of instances where this is the case.). Within a feature and attribute set for a function, if there is a feature named X then there will not be an attribute named X in the same function and vice versa in order to reduce confusion.

All features and attributes have two meta-data properties that define their usage within a function. Features and attributes can either be mandatory or optional; that is they must be provided or they can be provided if so desired. Optional features and attributes have a default value or a default meaning. When a default value is assumed the user will be notified in the response.

Features and attributes can also be single-valued or multi-valued; that is they can be specified, at most, only once or can be specified more than once. To supply more than one value for a multi-valued feature or attribute, specify the feature or attribute again with a new value. The philosophy is that one feature or attribute should be used to specify one value.

1.8 Notes

Notes are a means of providing information back to the user that are distinct from an error. There are many reasons why this may be so as some notes are cursory in nature where others are more critical. A note might be provided to indicate what default values were used or why some action was performed that is not typical.

A note has three components: a name, a priority and the text of the note itself. The name of a note may be called something like "`Get Information Success`". The priority of a note indicates how important the note is and can have one of the following values ranging from the most important to the least important: `CRITICAL`, `WARNING`, `INFO` and `DEBUG`. The text of the note will contain the message.

Notes are provided in the `notes` field of the `responses` element, the `response` element and the `responseResult` element.

1.9 Errors

Errors can occur for many reasons, too many to list exhaustively. Some of the possible errors include incorrectly specified attributes or missing mandatory values.

An error can occur at the request level or the global requests level. If an error is encountered at the global requests level then all individual requests are not processed.

If an error is encountered at the request level there will be no partial response returned or any partial data saved to the database. Other individual requests in the same query will be immune to this error and will be processed as normal.

An error has two components: a code and the text of the error itself. The code of the error is a number that uniquely identifies the type of the error. The text of the error will contain the description of the error.

Errors are reported in the `error` field of the `responses` element or the `response` element. Also the `status` field within the same element will have the value of `ERROR`.

1.10 The NAMF Server and Multi-threaded Implementations

NAMF Webservices may be implemented using multi-threaded architecture. Because of this there is no guarantee that multiple requests submitted together in a requests element will be processed in the order specified.

This is most apparent if a notifyAddress request and a getNotifications request are submitted together. Despite the responses for all the requests being returned at the same time, the timing of the processing the getNotifications request may or may not return the notifyAddress data. If the intention is for the getNotifications request is to return the submitted notifyAddress data then two separate requests, with the user guaranteeing the order of the request needs to be made.

2 Global Features

Feature Name	Occurrence	Optional
batchMode	Single-valued	Optional – Default: block Can be set to block, asynchronous or fetch. See the section on asynchronous processing later in this document.

3 Information Functions

The functions, **getInformation**, **getFieldNames** and **getFieldValues** form a set of online documentation functions to support the rest of the NAMF Web Service. This documentation reflects the actual running instance of the NAMF Server with its various versions.

3.1 getInformation Function

The **getInformation** function allows a vendor to retrieve information about all the functions and their various versions implemented on the NAMF Web Service. This online documentation reflects what is running on the NAMF Web Service.

IMPORTANT

It is very important that the output format of **getInformation** complies with the standard described below as the NAMF Compliance Tester relies on it being in this format.

To call the **getInformation** function the name field in the request must have the value of `getInformation`.

3.1.1 Request - Features & Attributes

The only input to the **getInformation** function is an optional feature. This feature has a default value if the feature is not supplied.

Feature Name	Occurrence	Optional
version	MULTIPLE	TRUE

Table 1 listAddress Request Feature Summary

The **getInformation** function can have a version feature. The version feature is used to retrieve information about a particular version. It can occur multiple times in order to specify multiple versions of interest. If it is not supplied the default version is the latest version. The version feature is specified with the name of `version` and the value being one of the version values in Table 2 Version ValuesTable 2 below.

version Value	Description
ALL	Get information about all functions for all versions. If specified it overrides all other

	values of the `version` feature.
`LATEST`	Get information about all the functions for the latest version.
version string	Get information about all functions for this version.

Table 2 Version Values

3.1.2 Response – Call Versions

The response of the **getInformation** function is unique in that it returns its response via the `callVersions` element. This is the only function to do so. The `callVersions` element provides a means of returning documentation about versions of a NAMF Web Service function. The `callVersions` element has a name and a list of call version.

The name (`name` element) is used to identify which function the `callVersions` element is describing. It has a value being one of the function names in Table 3 below.

name Value	Description
`getInformation`	The **getInformation** function name.
`getFieldNames`	The **getFieldNames** function name.
`getFieldValues`	The **getFieldValues** function name.
`notifyAddress`	The **notifyAddress** function name.
`getNotifications`	The **getNotifications** function name.
`validateAddress`	The **validateAddress** function name.
`listAddress`	The **listAddress** function name.
`getServerInformation`	This is where global features supported by the webservice are reported.

Table 3 Name Values

The call version (`callVersion` element) contains the documentation for a particular version of the named NAMF Web Service function. It contains a description, a version, a list of features and a list of attributes.

The description (`description` element of `callVersion`) contains a brief description of the function. This description is brief and is just enough to typify the purpose of the function. This document should be the source for definitive technical information.

The version (`version` element of `callVersion`) is the version of the function that this `callVersion` object is describing. In a list of `callVersion` objects, information about a version will appear only once.

The list of features (`features` element of `callVersion`) contains details about all the features that the function uses. This list could be empty if there are no features defined for the function. See the **Call Version – Feature and Attribute Metadata** section for information about how to interpret the features' metadata.

The list of attributes (`attributes` element of `callVersion`) contains details about all the attributes that the function uses. This list could be empty if there are no attributes defined for the function. See the **Call Version – Feature and Attribute Metadata** section for information about how to interpret the attributes' metadata.

```
<xs:element name="callVersions">
  <xs:complexType>
    <xs:sequence>
      <xs:element maxOccurs="unbounded" name="callVersion">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="description" type="xs:string" />
            <xs:element name="version" type="xs:string" />
            <xs:element minOccurs="0" ref="features" />
                  <xs:element minOccurs="0" ref="attributes" />
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
          <xs:attribute name="name" type="xs:string" />
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

XML Schema extract for the callVersions element from the NAMF XSD.

3.1.3 Call Version – Feature and Attribute Metadata

The information returned in the lists of features and attributes is metadata that describes each feature or attribute that is defined for the function that this `callVersion` object describes. This metadata information is common between features and attributes.

3.1.3.1 Features

Each feature is described in the following way:

`name` attribute	The name of the feature (e.g. `batchMode`)
`featureValue` element	A description of the feature (e.g. `The batch mode of the server.`)
Sub-feature with name `optional`	TRUE or FALSE
Sub-feature with name `occurrence`	`SINGULAR` or `MULTIPLE`
Sub-feature with name `default`	Default value (if has one)
Sub-feature with name `values`	Used to restrict the permitted values for this feature. A list of sub-features where name is `VALUE` or `LOOKUP`. If `VALUE` then the `featureValue` is the value, else if `LOOKUP` then `featureValue` is the value passed to the **getFieldValues** function to retrieve the list of permitted values.

Example Features Output:

```
<features>
	<feature name="version">
		<featureValue>The version(s) to get information about.</featureValue>
		<feature name="optional">
			<featureValue>TRUE</featureValue>
		</feature>
		<feature name="occurrence">
			<featureValue>MULTIPLE</featureValue>
		</feature>
		<feature name="default">
			<featureValue>LATEST</featureValue>
		</feature>
		<feature name="values">
			<feature name="LOOKUP">
				<featureValue>version</featureValue>
			</feature>
			<feature name="VALUE">
				<featureValue>LATEST</featureValue>
			</feature>
			<feature name="VALUE">
				<featureValue>ALL</featureValue>
			</feature>
		</feature>
	</feature>
</features>
```

3.1.3.2 Attributes

Attributes are described in a similar way to features. Each attributes is described in the following way:

`name` attribute	The name of the attribute (e.g. `notificationType`)
`attributeValue` element	A description of the feature (e.g. `The batch mode of the server.`)
Sub-attribute with name `optional`	TRUE or FALSE
Sub- attribute with name `occurrence`	`SINGULAR` or `MULTIPLE`
Sub- attribute with name `default`	Default value (if has one)
Sub- attribute with name `values`	Used to restrict the permitted values for this attribute. A list of sub-features where name is `VALUE` or `LOOKUP`. If `VALUE` then the `attributeValue` is the value, else if `LOOKUP` then `attributeValue` is the value passed to the **getFieldValues** function to retrieve the list of permitted values.

Example Attributes Output:

```
<attribute name="contributor">
	<attributeValue>The name of a Contributor to filter the result set with. Ignored if
not supplied.</attributeValue>
	<attribute name="optional">
```

```
                <attributeValue>TRUE</attributeValue>
        </attribute>
        <attribute name="occurrence">
                <attributeValue>MULTIPLE</attributeValue>
        </attribute>
        <attribute name="values">
                <attribute name="LOOKUP">
                        <attributeValue>contributor</attributeValue>
                </attribute>
                <attribute name="VALUE">
                        <attributeValue>ALL</attributeValue>
                </attribute>
                <attribute name="VALUE">
                        <attributeValue>ME</attributeValue>
                </attribute>
                <attribute name="VALUE">
                        <attributeValue>MY_ORGANISATION</attributeValue>
                </attribute>
        </attribute>
</attribute>
```

3.1.4 Example

Request:

```
<?xml version="1.0" encoding="utf-8"?>
<ns2:execute xmlns:ns2="http://ws.namf09.anzlic.org.au">
                        <requests xmlns="http://namf09.anzlic.org.au" id="100" version="0.2">
                                <authentication>
                                        <username>bwalker</username>
                                        <password>secret</password>
                                </authentication>
                                <features />
                                <request id="100.1" name="getInformation">
                                        <features />
                                </request>
                        </requests>
</ns2:execute>
```

Response - See **8 Appendix – Full getInformation Response XML Example** for the full response output.

3.2 getFieldNames Function

The **getFieldNames** function allows a vendor to retrieve a set of all the field names that have a dynamic set of values. The values of these fields are not able to be specified in the WSDL as the values will change over time. This function works in conjunction with the **getFieldValues** function which returns the set of values for a specified field name.

To call the **getFieldNames** function the name field in the request must have the value of `getFieldNames`.

3.2.1 Request - Features & Attributes

The **getFieldNames** function takes no features or attributes.

3.2.2 Response - Attributes

Attribute Name	Single-valued / Multi-valued	Mandatory / Optional
`fieldName`	Multi-valued	Mandatory (at least one)

Table 4 listAddress Request Response Attribute Summary

3.2.3 Example

Request:

```
<?xml version="1.0" encoding="utf-8"?>
<ns2:execute xmlns:ns2="http://ws.namf09.anzlic.org.au">
                        <requests xmlns="http://namf09.anzlic.org.au" id="100" version="0.2">
                                <authentication>
                                        <username>bcook</username>
                                        <password>secret</password>
                                </authentication>
                                <features />
                                <request id="100.1" name="getFieldNames">
                                        <features />
                                </request>
                        </requests>
</ns2:execute>
```

Response:

```
<ns2:executeResponse xmlns="http://namf09.anzlic.org.au"
xmlns:ns2="http://ws.namf09.anzlic.org.au">
        <responses id="100">
                <result hasErrorsInResponseElements="false" completed="true" status="OK" />
                <response id="100.1">
                        <responseResult>
                                <attributes>
                                        <attribute name="fieldName">
                                                <attributeValue>addressStatus</attributeValue>
                                        </attribute>
                                        <attribute name="fieldName">
                                                <attributeValue>areaOfInterest</attributeValue>
                                        </attribute>
                                        <attribute name="fieldName">
                                                <attributeValue>certainty</attributeValue>
                                        </attribute>
                                        <attribute name="fieldName">
                                                <attributeValue>complexLevelType</attributeValue>
```

```
                                    </attribute>
                                    <attribute name="fieldName">
                                            <attributeValue>complexUnitType</attributeValue>
                                    </attribute>
                                    <attribute name="fieldName">
                                            <attributeValue>contributor</attributeValue>
                                    </attribute>
                                    <attribute name="fieldName">
                                            <attributeValue>countryNameCode</attributeValue>
                                    </attribute>
                                    <attribute name="fieldName">
                                            <attributeValue>notificationType</attributeValue>
                                    </attribute>
                                    <attribute name="fieldName">

        <attributeValue>postalDeliveryTypeCode</attributeValue>
                                    </attribute>
                                    <attribute name="fieldName">
                                            <attributeValue>stateTerritory</attributeValue>
                                    </attribute>
                                    <attribute name="fieldName">
                                            <attributeValue>streetSuffix</attributeValue>
                                    </attribute>
                                    <attribute name="fieldName">
                                            <attributeValue>streetType</attributeValue>
                                    </attribute>
                                    <attribute name="fieldName">
                                            <attributeValue>version</attributeValue>
                                    </attribute>
                                </attributes>
                            </responseResult>
                            <status>OK</status>
                    </response>
            </responses>
</ns2:executeResponse>
```

3.3 GetFieldValues Function

The **getFieldValues** function allows a vendor to retrieve all the values for a specified field name. These values are a dynamic set of values for the specified field name that are not able to be specified in the WSDL. This function works in conjunction with the **getFieldNames** function which returns a list of field names.

To call the **getFieldValues** function the name field in the request must have the value of `getFieldValues`.

3.3.1 Request - Features & Attributes

Feature Name	Single-valued / Multi-valued	Mandatory / Optional
`fieldName`	Single-valued	Mandatory Must be a valid fieldname. i.e. one of the field names returned by a getFieldNames call.

Table 5 listAddress Request Feature Summary

No attributes are passed into the getFieldValues call.

3.3.2 Response - Attributes

Attribute Name	Single-valued / Multi-valued	Mandatory / Optional
`fieldValue`	Multi-valued	Mandatory

Table 6 listAddress Request Response Attribute Summary

3.3.3 Example

Request:

```
<ns2:execute xmlns:ns2="http://ws.namf09.anzlic.org.au">
	<requests xmlns="http://namf09.anzlic.org.au" id="100" version="0.2">
		<authentication>
			<username>bcook</username>
			<password>secret</password>
		</authentication>
		<features />
		<request id="100.1" name="getFieldValues">
			<features>
				<feature name="fieldName">
					<featureValue>contributor</featureValue>
				</feature>
			</features>
		</request>
	</requests>
</ns2:execute>
```

Response:

```
<ns2:executeResponse xmlns="http://namf09.anzlic.org.au"
xmlns:ns2="http://ws.namf09.anzlic.org.au">
```

```
        <responses id="100">
                <result hasErrorsInResponseElements="false" completed="true" status="OK" />
                <response id="100.1">
                        <responseResult>
                                <attributes>
                                        <attribute name="fieldValue">
                                                <attributeValue>Acme Pty Ltd</attributeValue>
                                        </attribute>
                                        <attribute name="fieldValue">
                                                <attributeValue>PSMA</attributeValue>
                                        </attribute>
                                </attributes>
                        </responseResult>
                        <status>OK</status>
                </response>
        </responses>
</ns2:executeResponse>
```

4 Notification Functions

4.1 notifyAddress Function

The **notifyAddress** function allows a vendor to submit a notification about a new address, about an incorrect address, to modify an existing address, about a deleted address, about a new address alias or to comment an existing address.

To call the **notifyAddress** function the name field in the request must have the value of `notifyAddress`.

4.1.1 Request - Features & Attributes

The **notifyAddress** function requires an address and a list of attributes. No features are supported. The address is mandatory. Some attributes are mandatory and some attributes are optional. All attributes are single-valued. All optional attributes have a default value.

The complete input requirements are highly dependent upon the notification type attribute. The remainder of this section outlines the complete list of allowed attributes and their usage; and the links between some attributes and the notification type attribute.

Attribute Name	Single-valued / Multi-valued	Mandatory / Optional
`notificationType`	Single-valued	Mandatory
`certainty`	Single-valued	Mandatory
`effectiveDate`	Single-valued	Optional – Default: current date
`aliasAddressDetailPID`	Single-valued	Mandatory if notificationType is `I_ALIAS` otherwise it is ignored
`comments`	Single-valued	Mandatory if notificationType is `X_COMMENT` otherwise it is ignored.
`areaOfInterest`	Multi-valued	Mandatory if notifying an unstructured address. Optional if notifying a structured address.

Table 7 notifyAddress Attribute Summary

notificationType Value	Description
`I_INSERT`	The address does not exist in the reference dataset.
`I_ALIAS`	The address is an alias for an existing address in the reference dataset.
`U_ATTRIB_GEO`	Both the attribute and the geocode information require an update.
`U_ATTRIB_ONLY`	Only the attribute information requires an update.
`U_GEO_ONLY`	Only the geocode information requires an update.
`X_COMMENT`	The address exists in the reference dataset. The notifier wishes to add a comment.
`D_GREENSPACE`	The address is a walkway, park, or reserve and can be deleted.
`D_ROAD_RESERVE`	The address corresponds to a road reserve and can be deleted.
`D_NOT_EXISTS`	The address either no longer exists, or never existed and can be deleted.

Table 8 notifyAddress Notification Type Values

The **notifyAddress** function has 9 distinct notification types. The notification type is a single-valued mandatory attribute that needs to be supplied for a **notifyAddress** function to succeed. The notification type attribute is specified with the name of `notificationType` and the value being one of the notificationType values in Table 8.

A notification sent via the **notifyAddress** function also must has a certainty. The certainty is a single-valued mandatory attribute that needs to be supplied for a **notifyAddress** function to succeed. The certainty attribute is specified with the name of `certainty` and the value being one of the certainty values in Table 9

Certainty Value	Description
`full`	Fully certain that the notification is valid (90 - 100%).
`partial`	Partially certain that the notification is valid (< 90%).

Table 9 Certainty Values

A notification can have an effective date attribute. The effective date is a single-valued optional attribute which defaults to the current date if not supplied. The effective date attribute is specified with the name of `effectiveDate`. The format of the effective date is 'dd/mm/yy' where dd, mm or yy is a two digit day, month or year respectively.

For an alias notification (notificationType is `I_ALIAS`) the alias address detail PID must be supplied. The alias address detail PID is a single-value mandatory attribute for this notification type and needs to be supplied for an alias notification to succeed. If specified for other notification types it is ignored. The alias address detail PID attribute is specified with the name of `aliasAddressDetailPID`. The value of an address detail PID attribute should be a valid PAF identifier or a valid GNAF identifier.

For a comment notification (notificationType is `X_COMMENT`) the comment must be supplied. The comment is a single-valued mandatory attribute for this notification type and needs to be supplied for a comment notification to succeed. If specified for other notification types it is ignored. The comment attribute is specified with the name of `comments`. The value of a comment attribute is a string up to 500 characters in length.

Accompanying the list of notification attributes must be a valid address. The minimum requirements for a valid address (for processing the address within the **notifyAddress** function only) are the following:

- Strictly **EITHER** the unstructured address details (the four address fields prefixed with `unstructured`) must be populated **OR** the structured address details (the remaining address fields excluding those prefixed with `geo`) must be populated. Populated means that some element in the set of fields must be a non-empty string.

If the address is involved in a geocode notification (notificationType is `U_ATTRIB_GEO` or `U_GEO_ONLY`) the geocode details (the address fields prefixed with `geo`) must be populated. Populated has the same meaning as above.

If the address is involved in an insert notification (notificationType is `I_INSERT` or `I_ALIAS`) then the `addressIdentifier` field in the address must be undefined (empty). An address identifier is generated when the notification has been validated and the address has been accepted as part of the reference dataset.

If the address is involved in a non-insert notification (notificationType is neither `I_INSERT` nor `I_ALIAS`) then the `addressIdentifier` field in the address must be defined (populated). This is because the notification is about an existing address in a reference dataset.

4.1.2 Response - Attributes & Notes

Attribute Name	Single-valued / Multi-valued	Mandatory / Optional
`notificationID`	Single-valued	Mandatory – the internal identifier assigned to the notification
input attributes		The input attributes provided are repeated back in the responseResult object.

Table 10 notifyAddress Response Attribute Summary

All the used input attributes are copied to the list of output attributes.

When a notification has been successful a notification ID attribute is also returned. The notification ID attribute is specified with the name of `notificationID` and the value being a positive integer identifier. This notification ID uniquely identifies the notification within the NAMF Web Service.

The successful submission of a notification as signified by the return of a notification ID does not mean that the notification is valid and has been accepted. In order for the NAMF Web Service to be responsive, the validation of notifications is done offline as it is a time-consuming operation.

Note Name	Single-valued / Multi-valued	Mandatory / Optional
`effectiveDateDefault`	Single-valued	Optional – provided if the effective date attribute for the notification was not provided

Table 11 notifyAddress Response Note Summary

4.1.3 Example

Request:

```
<ns2:execute xmlns:ns2="http://ws.namf09.anzlic.org.au">
        <requests xmlns="http://namf09.anzlic.org.au" id="100" version="0.2">
                <authentication>
                        <username>bcook</username>
                        <password>secret</password>
                </authentication>
                <features />
                <request id="100.1" name="notifyAddress">
                        <features />
                        <address>
```

```
                        <addressIdentifier>GANT_111222333</addressIdentifier>
                        <streetNumber1>100</streetNumber1>
                        <streetName>Sesame</streetName>
                        <streetType>ST</streetType>
                        <streetSuffix>
                        </streetSuffix>
                        <localityName>DARWIN</localityName>
                        <stateTerritory>NT</stateTerritory>
                </address>
                <attributes>
                        <attribute name="notificationType">
                                <attributeValue>D_GREENSPACE</attributeValue>
                        </attribute>
                        <attribute name="certainty">
                                <attributeValue>FULL</attributeValue>
                        </attribute>
                </attributes>
        </request>
    </requests>
</ns2:execute>
```

Response:

```
<?xml version="1.0" encoding="utf-8"?>
<ns2:executeResponse xmlns="http://namf09.anzlic.org.au"
xmlns:ns2="http://ws.namf09.anzlic.org.au">
    <responses id="100">
        <result hasErrorsInResponseElements="false" completed="true" status="OK" />
        <response id="100.1">
                <responseResult>
                        <address>
                                <addressIdentifier>GANT_111222333</addressIdentifier>
                                <streetNumber1>100</streetNumber1>
                                <streetName>Sesame</streetName>
                                <streetType>ST</streetType>
                                <streetSuffix>
                                </streetSuffix>
                                <localityName>DARWIN</localityName>
                                <stateTerritory>NT</stateTerritory>
                        </address>
                        <attributes>
                                <attribute name="notificationType">
                                        <attributeValue>D_GREENSPACE</attributeValue>
                                </attribute>
                                <attribute name="certainty">
                                        <attributeValue>FULL</attributeValue>
                                </attribute>
                                <attribute name="effectiveDate">
                                        <attributeValue>16/06/2010</attributeValue>
                                </attribute>
                                <attribute name="notificationID">
                                        <attributeValue>11966</attributeValue>
                                </attribute>
                        </attributes>
                        <notes>
                                <note priority="INFO" name="effectiveDateDefault">The
effective date attribute has defaulted to the current date.</note>
                        </notes>
                </responseResult>
```

```
                    <status>OK</status>
                </response>
            </responses>
</ns2:executeResponse>
```

4.2 getNotifications Function

The **getNotifications** function allows a vendor to query the set of notifications with a set of search criteria.

To call the **getNotifications** function the name field in the request must have the value of `getNotifications`.

4.2.1 Request - Features & Attributes

The **getNotifications** function can accept a list of features and attributes. All features are single-valued and optional with default values. All attributes are optional with default values. Some attributes are single-valued and some attributes are multi-valued.

Feature Name	Single-valued / Multi-valued	Mandatory / Optional
`markAsRead`	Single-valued	Optional – Default: `false`
`maxResults`	Single-valued	Optional – Default: `50`

Table 12 getNotifications Request Feature Summary

Attribute Name	Single-valued / Multi-valued	Mandatory / Optional
`startDate`	Single-valued	Optional – Default: ignored
`endDate`	Single-valued	Optional – Default: ignored
`contributor`	Multi-valued	Optional – Default: user's subscription
`notificationType`	Multi-valued	Optional – Default: ignored
`ignoreReadStatus`	Single-valued	Optional – Default: `false`
`duplicateHandling`	Single-valued	Optional – Default: NONE

Table 13 getNotifications Request Attribute Summary

A **getNotifications** request can have a mark as read feature. The mark as read feature is a single-value optional feature which defaults to `false` if not supplied. The mark as read feature is specified with the name of `markAsRead` and can have a value of `true` or `false`. This feature indicates whether to automatically mark all downloaded notifications as read.

A **getNotifications** request can have a maximum results feature. The maximum results feature is a single-valued optional feature which defaults to `50` if not supplied. The maximum result feature is specified with the name of `maxResults` and can have a positive integer value less than `1000`. Values greater than `1000` will revert to `1000`. This feature indicates the maximum number of notifications to receive.

A **getNotifications** request can have a start date attribute. The start date attribute is a single-valued optional attribute which is ignored if not supplied. The start date attribute is specified with the name of `startDate`. This attribute indicates the start of a date range for which to return notifications created on or after this date.

A **getNotifications** request can have an end date attribute. The end date attribute is a single-valued optional attribute which is ignored if not supplied. The end date attribute is specified with the name of `endDate`. This attribute indicates the end of a date range for which to return notifications created on or before this date.

The date format for a start date attribute, or an end date attribute can be specified using a child attribute with the name of `format`. The default value for this date format attribute is `DD/MM/YYYY`. The date format specifies, in any order, the day, the month and the year separated by a forward slash or a dash. See the following tables and examples for more information.

Specification	Meaning	Examples
DD	Day of the month as an integer.	01, 21
MM	Month of the year as an integer.	02, 11
MMM	Month of the year as an abbreviation.	Mar, Sep
YY	Year in the current century.	12 meaning 2012
YYYY	Year.	2011

Table 14 Date format specification

Date	Format
06/04/2010	DD/MM/YYYY
JAN-06-10	MMM-DD-YY
2011-08-31	YYYY-MM-DD

Table 15 Date format examples

A **getNotifications** request can have a contributor attribute. The contributor attribute is a multi-valued optional attribute with the default of applying the user's subscription if not supplied. The contributor attribute is specified with the name of `contributor` and the value being one of the contributor values in Table 16. This attribute indicates the list of contributors for which to return their notifications.

contributor Value	Description
`ALL`	Do not filter using the contributor attribute. It must be the only contributor attribute specified. An error is generated if any other contributor attribute is specified.
`ME`	Retrieve all notifications the current user has submitted. The result set will contain valid, invalid and yet to be validated notifications. This attribute can not be used with any other Contributor attribute. An error is generated if any other contributor attribute is specified.
`MY_ORGANISATION`	Retrieve all notifications that have been submitted by the current users organisation. The result set will contain valid, invalid and yet to be validated notifications. This attribute can not be used with any other Contributor attribute. An error is generated if any other contributor attribute is specified.
Contributor Name Eg. "PSMA"	Retrieve all notifications from the supplying organisation. The result set will only contain valid notifications. It can be specified along with other contributor attributes that specify actual contributors (cannot be `ALL`). The list of supplying organisations is not documented here. An example might be 'AEC-NSW' for the Australian Electoral Commission of New South Wales.

No Contributor Attribute Specified	Retrieve all valid notifications for the organisations defined in the current users subscription.

Table 16 Contributor Values

A **getNotifications** request can have a notification type attribute. The notification type is a multi-valued optional attribute which is ignored if not supplied. The notification type attribute is specified with the name of `notificationType` and the value being one of the notificationType values in Table 8. This attribute indicates the type of notifications to return.

A **getNotifications** request can have an ignore-read status attribute. The ignore-read status attribute is a single-valued optional attribute which if not specified defaults to false. The ignore-read status attribute is specified with the name of `ignoreReadStatus` and can have a value of `true` or `false`. This attribute indicates whether to ignore the read status of previous read notifications.

A **getNotifications** request can have a duplicate handling attribute. The duplicate handling attribute is a single-valued optional attribute with the default of NONE. The duplicate handling attribute is specified with the name of `duplicateHandling` and the value being one of the values in Table 17.

contributor Value	Description
`NONE`	No duplicates will be returned in the response. Note that the notifications are only removed from a response if they are a duplicate of another notification within the same response. The earliest notification is returned in preference over others.
`ALL`	No restrictions. Duplicates can be returned.
`IGNORE_SAME_AS_MY_ORG`	Will only return duplicates if they do not match a notification submitted by the user's organisation. This allows a user to totally ignore all notifications in the system that match their own organisation's notifications.

Table 17 Duplicate Handling Values

Consider the following scenario where 3 notifications exist in the NCS system.

Contributor Organisation	Notification ID	Duplicate of Notification ID
A	1	
B	2	1
C	3	1

Now consider the case where a user from Organisation A performs a search for all notifications from Organisation B or C. The table below details which notifications they would receive for different duplicate handling settings.

Duplicate Handling Value	Notifications Returned
`NONE`	2
`ALL`	1,2 & 3
`IGNORE_SAME_AS_MY_ORG`	none

Table 18 Duplicate Handling Example

4.2.2 Response - Attributes & Notes

Attribute Name	Single-valued / Multi-valued	Mandatory / Optional
notificationID	Single-valued	Mandatory – the internal identifier assigned to the notification
notificationDate	Single-valued	Mandatory – date the notification was received. Format dd/mm/yyyy
effectiveDate	Single-valued	Mandatory – date the notification becomes effective. Format dd/mm/yyyy
contributor	Single -valued	Optional – the name of the organisation that contributed this notification. Will not be provided if this notification came from a user who was not associated with an organisation.
notificationType	Single -valued	Mandatory
aliasAddressDetailPID	Single -valued	Mandatory if notificationType is I_ALIAS.
comments	Single -valued	Mandatory if notificationType is X_COMMENT.

duplicateOfNotification	Multi-Valued	Optional – only populated if this notification is a duplicate of other notification(s). A complex structured attribute which contains information about the notification ID and contributor of the duplicate notification. E.g.:
		
		`<attribute name="duplicateOfNotification">`
		`<attribute name="notificationID">`
		`<attributeValue>345345</attributeValue>`
		`</attribute>`
		`<attribute name="contributor">`
		`<attributeValue>Acme Pty Ltd`
		`</attributeValue>`
		`</attribute>`
		`</attribute>`

Table 19 getNotifications Response Attribute Summary

Note Name	Single-valued / Multi-valued	Mandatory / Optional
resultCount	Single-valued	Mandatory – the number of notifications returned
potentialResultCount	Single-valued	Optional – how many notifications matched the search criteria. May be more than 'resultCount' if the number of matching notifications was more than the maximum number of notifications allowed to be returned.
numberOfNotifications ReceivedForTheFirstTime	Single-valued	Mandatory – The number of notifications retrieved for the first

		time (by the calling user). Will be less than or equal to 'resultCount'.
numberOfNotifications MarkedAsRead	Single -valued	Mandatory – The number of notifications already marked as 'read' for the calling user. Will be less than or equal to 'resultCount'.
numberOfNotifications AlreadyMarkedAsRead	Single -valued	Mandatory – The number of notifications marked as 'read' for the calling user. Will be less than or equal to 'resultCount'.

Table 20 getNotifications Response Note Summary

4.2.3 Example

Request:

```
<?xml version="1.0" encoding="utf-8"?>
<ns2:execute xmlns:ns2="http://ws.namf09.anzlic.org.au">
	<requests xmlns="http://namf09.anzlic.org.au" id="100" version="0.3">
		<authentication>
			<username>amahar</username>
			<password>amahar</password>
		</authentication>
		<features />
		<request id="100.3" name="getNotifications">
			<features>
				<feature name="maxResults">
					<featureValue>2</featureValue>
				</feature>
			</features>
			<attributes>
				<attribute name="contributor">
					<attributeValue>PSMA</attributeValue>
				</attribute>
				<attribute name="ignoreReadStatus">
					<attributeValue>true</attributeValue>
				</attribute>
			</attributes>
		</request>
	</requests>
</ns2:execute>
```

Response:

```
<?xml version="1.0" encoding="utf-8"?>
<ns2:executeResponse xmlns="http://namf09.anzlic.org.au"
xmlns:ns2="http://ws.namf09.anzlic.org.au">
	<responses id="100">
```

```
<result hasErrorsInResponseElements="false" completed="true" status="OK" />
<response id="100.3">
	<responseResult>
		<address>
			<addressIdentifier>5349</addressIdentifier>
			<streetNumber1>55A</streetNumber1>
			<streetName>Marylou</streetName>
			<streetType>CNR</streetType>
			<streetSuffix>S</streetSuffix>
			<localityName>Risdon Vale</localityName>
			<stateTerritory>TAS</stateTerritory>
		</address>
		<attributes>
			<attribute name="contributor">
				<attributeValue>PSMA</attributeValue>
			</attribute>
			<attribute name="notificationID">
				<attributeValue>5346</attributeValue>
			</attribute>
			<attribute name="notificationDate">
				<attributeValue>03/05/2010</attributeValue>
			</attribute>
			<attribute name="certainty">
				<attributeValue>Full</attributeValue>
			</attribute>
			<attribute name="effectiveDate">
				<attributeValue>03/05/2010</attributeValue>
			</attribute>
			<attribute name="notificationType">
				<attributeValue>I_INSERT</attributeValue>
			</attribute>
		</attributes>
	</responseResult>
	<responseResult>
		<address>
			<addressIdentifier>5360</addressIdentifier>
			<status>OFF</status>
			<streetNumber1>25</streetNumber1>
			<streetName>Ferry</streetName>
			<streetType>RD</streetType>
			<localityName>Lindisfarne</localityName>
			<stateTerritory>TAS</stateTerritory>
			<postcode>7015</postcode>
		</address>
		<attributes>
			<attribute name="contributor">
				<attributeValue>PSMA</attributeValue>
			</attribute>
			<attribute name="notificationID">
				<attributeValue>5357</attributeValue>
			</attribute>
			<attribute name="notificationDate">
				<attributeValue>04/05/2010</attributeValue>
			</attribute>
			<attribute name="certainty">
				<attributeValue>Full</attributeValue>
			</attribute>
			<attribute name="effectiveDate">
				<attributeValue>04/05/2010</attributeValue>
			</attribute>
			<attribute name="notificationType">
				<attributeValue>I_INSERT</attributeValue>
			</attribute>
		</attributes>
```

```
                        </responseResult>
                        <notes>
                                <note priority="WARNING" name="restriction">These results have
been restricted to your area of interests which are ACT, NSW, NT, OT, QLD, SA, TAS, VIC,
WA</note>
                                <note priority="INFO" name="resultCount">2</note>
                                <note priority="INFO" name="potentialResultCount">227</note>
                                <note priority="INFO"
name="numberOfNotificationsReceivedForTheFirstTime">0</note>
                                <note priority="INFO"
name="numberOfNotificationsAlreadyMarkedAsRead">2</note>
                                <note priority="INFO"
name="numberOfNotificationsMarkedAsRead">0</note>
                        </notes>
                        <status>OK</status>
                </response>
        </responses>
</ns2:executeResponse>
```

5 Address Functions

5.1 validateAddress Function

The validateAddress function can be used to:

- Confirm that an address is in the authoritative dataset (Full matching)
- Suggest alternative authoritive addresses if a full match cannot be found for a candidate address (Partial matching)
- Provide geocode details for an address
- Perform reverse-geocoding (get the address closest to a point)

5.1.1 Request - Features & Attributes & Address

	Feature Name	Single-valued / Multi-valued	Mandatory / Optional
	maxResults	Single-valued	Optional – Default: Unlimited Only useful when doing 'Partial' match. e.g. if set to 3 then return the best 3 candidates. If more than maxResults addresses match then limit the results returned to the best 3 as determined by the engine. NOTE: if minMatchingAccuracy is set to 'Full' and there is more than 1 address that matches, return an error even if maxResults is 1.
Features that control how the address search is performed	minMatchingAccuracy	Single-valued	Optional – Default 'full' "full" or "partial" Only applicable to structured and unstructured addresses – otherwise ignored.
	validationType	Single-valued	Optional – Default: structuredAddress - structuredAddress - unstructuredAddress - addressIdentifier - cadastralIdentifier - reverseGeocode Tells the server what type of address is being passed in (and thus what type

			of search should be performed). A server may support 1 or many of these types. The values supported can be found from the getInformation request. See the section on Input Address Types below for more information.
Features that control what address fields are returned	geocode	Single-valued	Optional – Default: false If set to true the geo* fields of the address are returned in the response.
	omitAddress	Single-valued	Optional – Default: false If set to true the address line fields of the address are NOT returned in the response. If this is set to true then geocode must be set to true, otherwise the request makes no sense.
	getPostalAddress	Single-valued	Optional – Default: false If set to true then the postalDeliveryTypeCode and postalDeliveryIdentifier fields are returned.

Table 21 validateAddress Request Feature Summary

There are no attributes for the validateAddress request.

5.1.2 Input Address Types

The input address fields that need to be provided depends on the validationType specified:

structuredAddress	Only structured address fields can be populated
unstructuredAddress	Only unstructured address fields can be populated
addressIdentifier	Only addressIdentifier field can be populated
cadastralIdentifier	Only cadastralIdentifier field can be populated
reverseGeocode	Only the geoNorthSouthCoordinate & geoEastWestCoordinate fields and optionally the geoDatumCode field must be populated.

5.1.3 Response - Attributes & Notes

Attribute Name	Single-valued / Multi-valued	Mandatory / Optional
matchCertainty	Single-valued	Mandatory - full or partial
matchCertaintyMessage	Multi-valued	Optional Comments on the matching result – e.g. "Partial match on Street Number"

Table 22 validateAddress Request Response Attribute Summary

5.1.4 Matching Rules

There are different rules for matching depending on the validationType and minMatchingAccuracy provided.

validationType	**minMatching Accuracy**	**Matching Rules**
structuredAddress & unstructuredAddress	full	Every field that has been populated in the input address must match exactly (case-insensitive) to the authority address. Very strict. Null fields provided must be null in the response (except fields in the list below). Populated fields must match exactly (case-insensitive). Exceptions of things that can be null or different: - siteName (can be null or different) - postcode (can be null) - countryNameCode (can be null) - addressIdentifier (can be null on input) - cadastralIdentifier (can be null on input) - lotIdentifier can be null when streetNumber1 is supplied. - status - All geo* fields - locationDescriptor
	partial	No specific rules. It is up to the particular webservice to implement its own logic.
addressIdentifier	N/A	addressIdentifier must match exactly (case-sensitive)
cadastralIdentifier	N/A	cadastralIdentifier must match exactly (case-sensitive)
reverseGeocode	N/A	No specific rules. It is up to the server's discretion to determine how far away an address can be yet still be classed as a match.

5.1.5 What if no match is found?

If no match is found the response should contain no addresses.

A note called `addressFound` must be supplied with a value of false.

A note called `addressProblem` may be supplied however this is optional.

```
<?xml version="1.0" encoding="utf-8"?>
<ns2:executeResponse xmlns="http://namf09.anzlic.org.au"
xmlns:ns2="http://ws.namf09.anzlic.org.au">
        <responses id="120927">
                <result completed="true" status="OK" hasErrorsInResponseElements="false" />
                <response id="120927.1">
                        <notes>
                                <note priority="INFO" name="addressFound">false</note>
                                <note priority="INFO" name="addressProblem">Locality 'Lavender
Bay' does not exist</note>
                        </notes>
                        <status>OK</status>
                </response>
        </responses>
</ns2:executeResponse>
```

5.1.6 What if the matched address is an address-level alias?

If the address matched is an address-level alias information about its principal address must be provided in the notes section.

A note called `principalAddressDetailPid` must be supplied.

5.1.7 Example

Request:

```
<?xml version="1.0" encoding="utf-8"?>
<execute xmlns="http://ws.namf09.anzlic.org.au">
        <ns1:requests id="81643" version="0.3" xmlns:ns1="http://namf09.anzlic.org.au">
                <ns1:authentication>
                        <ns1:username>bwalker</ns1:username>
                        <ns1:password>secret</ns1:password>
                </ns1:authentication>
                <ns1:features />
                <ns1:request id="81643.1" name="validateAddress">
                        <ns1:features>
                                <ns1:feature name="validationType">
                                        <ns1:featureValue>unstructuredAddress</ns1:featureValue>
                                </ns1:feature>
                                <ns1:feature name="minMatchingAccuracy">
                                        <ns1:featureValue>Full</ns1:featureValue>
                                </ns1:feature>
                                <ns1:feature name="geocode">
                                        <ns1:featureValue>true</ns1:featureValue>
                                </ns1:feature>
                        </ns1:features>
                        <ns1:address>
                                <ns1:unstructuredAddressLine1>15A LITHGOW
STREET</ns1:unstructuredAddressLine1>
```

```
                        <ns1:unstructuredAddressLine2>FYSHWICK
ACT</ns1:unstructuredAddressLine2>
                    </ns1:address>
                    <ns1:attributes />
                </ns1:request>
        </ns1:requests>
</execute>
```

Response:

```
<?xml version="1.0" encoding="utf-8"?>
<ns2:executeResponse xmlns="http://namf09.anzlic.org.au"
xmlns:ns2="http://ws.namf09.anzlic.org.au">
        <responses id="81643">
                <result completed="true" status="OK" hasErrorsInResponseElements="false" />
                <response id="81643.1">
                        <responseResult>
                                <address>
                                        <addressIdentifier>GAACT717756877</addressIdentifier>
                                        <status>OFF</status>
                                        <streetNumber1>15A</streetNumber1>
                                        <streetName>LITHGOW</streetName>
                                        <streetType>STREET</streetType>
                                        <localityName>FYSHWICK</localityName>
                                        <stateTerritory>ACT</stateTerritory>
                                        <geoFeature>PROPERTY/PARCEL GEOCODE</geoFeature>
                                        <geoDatumCode>GDA94</geoDatumCode>
                                        <geoNorthSouthCoordinate>-
35.32923608</geoNorthSouthCoordinate>

        <geoEastWestCoordinate>149.16769176</geoEastWestCoordinate>
                                        <geoContainment>YES</geoContainment>
                                </address>
                                <attributes>
                                        <attribute name="matchCertainty">
                                                <attributeValue>Full</attributeValue>
                                        </attribute>
                                </attributes>
                        </responseResult>
                        <status>OK</status>
                </response>
        </responses>
</ns2:executeResponse>
```

5.2 listAddress Function

The listAddress function is used to support a user entering address details onto a form. It returns a list of possible values for a particular address field, based upon the address details the user has entered so far.

5.2.1 Request - Features & Attributes

Feature Name	Single-valued / Multi-valued	Mandatory / Optional
`maxResults`	Single-valued	Optional – Default: 50 Has hard maximum of 1000. That is, if a value over 1000 is specified the server will use a value of 1000.
`minMatchingAccuracy`	Single-valued	Optional - default “full”. “full” or “partial”. Partial matching allows the server to implement some sophisticated smarts. For example, a listAddress call with partial matching for street name of "SMI*" could return "SMYTH" through a 'sounds like' match clause.

Table 23 listAddress Request Feature Summary

Attribute Name	Single-valued / Multi-valued	Mandatory / Optional
`listField`	Single-valued	Mandatory A field name from the structured portion of the address structure. Not all structured address fields are supported. The allowed values can be found by performing a ‘getFieldValues’ request with a name of ‘listField’. The minimum fields required to be supported are: • `complexUnitType` • `complexLevelType` • `complexStreetName` • `complexStreetSuffix` • `streetName` • `streetType` • `streetSuffix` • `localityName` • `addressIdentifier`

Table 24 listAddress Request Attribute Summary

5.2.2 Input Address

The input address will be a partially populated structured address. The unstructured and geocode fields in the address are ignored. At least locality_name or state/territory must be populated. Only the field specified as `listField` may have wildcard characters (*). The wildcard '*' means 0 or many characters. Wildcards are not mandatory for the `listField` field. The `listField` in the address may be left blank. Searches are not case sensitive.

When 'domain' type fields are specified as `listField` the 'label' text should be searched on as well, not just the code value. For example, a search on "23 Smith Co*" could return "23 Smith Corner, Murraytown, NSW" even though the code for 'CORNER' is 'CNR'.

5.2.3 Response - Attributes & Notes

Attribute Name	Single-valued / Multi-valued	Mandatory / Optional
<result>	Multi-Valued	Optional. Name = the result Value = the code value (if this was a lookup to an enumerated field such as street type, street suffix, etc...), otherwise equal to 'name'

Table 25 listAddress Request Response Attribute Summary

Note Name	Single-valued / Multi-valued	Mandatory / Optional
`totalMatchesFound`	Single-valued	Optional Specified if there were more than `maxResults` matches. In this case it is likely the user has not specified enough filtering information and too many results have matched. The user should refine their address information and try again.
`maxResults`	Single-valued	Optional Specified if the `maxResults` feature was not provided (so used default) or if the value was overrided.

Table 26 listAddress Request Response Note Summary

5.2.4 Example 1

Request:

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<execute xmlns="http://ws.namf09.anzlic.org.au">
	<ns1:requests id="73934" version="0.3" xmlns:ns1="http://namf09.anzlic.org.au">
		<ns1:authentication>
			<ns1:username>bwalker</ns1:username>
			<ns1:password>secret</ns1:password>
		</ns1:authentication>
		<ns1:features />
		<ns1:request id="73934.1" name="listAddress">
			<ns1:features>
				<ns1:feature name="minMatchingAccuracy">
					<ns1:featureValue>full</ns1:featureValue>
				</ns1:feature>
			</ns1:features>
			<ns1:address>
				<ns1:streetName>MO*</ns1:streetName>
				<ns1:localityName>PARKES</ns1:localityName>
			</ns1:address>
			<ns1:attributes>
				<ns1:attribute name="listField">
					<ns1:attributeValue>streetName</ns1:attributeValue>
				</ns1:attribute>
			</ns1:attributes>
		</ns1:request>
	</ns1:requests>
</execute>
```

Response:

```
<?xml version="1.0" encoding="utf-8"?>
<ns2:executeResponse xmlns="http://namf09.anzlic.org.au"
xmlns:ns2="http://ws.namf09.anzlic.org.au">
	<responses id="68425">
		<result completed="true" status="OK" hasErrorsInResponseElements="false" />
		<response id="68425.1">
			<responseResult>
				<attributes>
					<attribute name="MOLONG" />
					<attribute name="MONASTRY" />
					<attribute name="MONICA" />
					<attribute name="MOON" />
					<attribute name="MOOR" />
					<attribute name="MOSSGIEL" />
					<attribute name="MOULDEN" />
				</attributes>
			</responseResult>
			<status>OK</status>
		</response>
	</responses>
</ns2:executeResponse>
```

5.2.5 Example 2

Request:

```
<?xml version="1.0" encoding="utf-8"?>
<execute xmlns="http://ws.namf09.anzlic.org.au">
	<ns1:requests id="110995" version="0.3" xmlns:ns1="http://namf09.anzlic.org.au">
		<ns1:authentication>
			<ns1:username>${USERNAME}</ns1:username>
			<ns1:password>${PASSWORD}</ns1:password>
```

```
                </ns1:authentication>
                <ns1:features />
                <ns1:request id="110995.1" name="listAddress">
                        <ns1:features />
                        <ns1:address>
                                <ns1:streetType>R*</ns1:streetType>
                                <ns1:localityName>MARNOO EAST</ns1:localityName>
                        </ns1:address>
                        <ns1:attributes>
                                <ns1:attribute name="listField">
                                        <ns1:attributeValue>streetType</ns1:attributeValue>
                                </ns1:attribute>
                        </ns1:attributes>
                </ns1:request>
        </ns1:requests>
</execute>
```

Response:

```
<?xml version="1.0" encoding="utf-8"?>
<ns2:executeResponse xmlns="http://namf09.anzlic.org.au"
xmlns:ns2="http://ws.namf09.anzlic.org.au">
        <responses id="110995">
                <result completed="true" status="OK" hasErrorsInResponseElements="false" />
                <response id="110995.1">
                        <responseResult>
                                <attributes>
                                        <attribute
name="RAMBLE"><attributeValue>RMBL</attributeValue></attribute>
                                        <attribute
name="RAMP"><attributeValue>RAMP</attributeValue></attribute>
                                        <attribute
name="RANAE"><attributeValue>RAN</attributeValue></attribute>
                                        <attribute
name="RANGE"><attributeValue>RNGE</attributeValue></attribute>
                                        <attribute
name="REACH"><attributeValue>RCH</attributeValue></attribute>
                                        <attribute
name="REEF"><attributeValue>REEF</attributeValue></attribute>
                                        <attribute
name="RESERVE"><attributeValue>RES</attributeValue></attribute>
                                        <attribute
name="ROAD"><attributeValue>RD</attributeValue></attribute>
                                        <attribute
name="ROADS"><attributeValue>RDS</attributeValue></attribute>
                                        <attribute
name="ROADSIDE"><attributeValue>RDSD</attributeValue></attribute>
                                        <attribute
name="ROADWAY"><attributeValue>RDWY</attributeValue></attribute>
                                        <attribute
name="RONDE"><attributeValue>RNDE</attributeValue></attribute>
                                        <attribute
name="ROSEBOWL"><attributeValue>RSBL</attributeValue></attribute>
                                        <attribute
name="ROTARY"><attributeValue>RTY</attributeValue></attribute>
                                        <attribute
name="ROUND"><attributeValue>RND</attributeValue></attribute>
                                        <attribute
name="ROUTE"><attributeValue>RTE</attributeValue></attribute>
                                        <attribute
name="ROW"><attributeValue>ROW</attributeValue></attribute>
                                        <attribute
```

```
name="ROWE"><attributeValue>ROWE</attributeValue></attribute>
                                        <attribute
name="RUA"><attributeValue>RUA</attributeValue></attribute>
                                        <attribute
name="RUE"><attributeValue>RUE</attributeValue></attribute>
                                        <attribute
name="RUN"><attributeValue>RUN</attributeValue></attribute>
                                </attributes>
                        </responseResult>
                        <status>OK</status>
                </response>
        </responses>
</ns2:executeResponse>
```

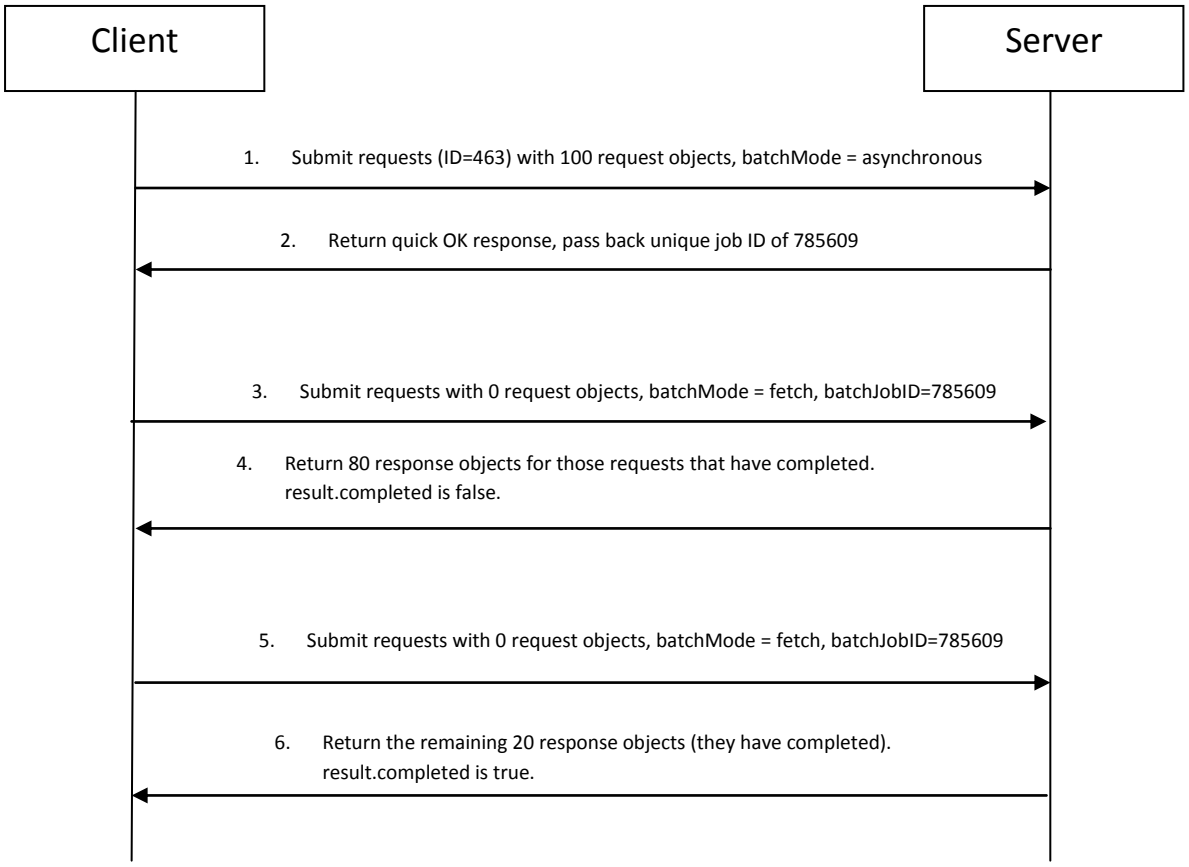
6 Asynchronous Processing

All calls may be performed in synchronous (default) mode. If the webservice supports asynchronous batchMode then all calls may also be performed in asynchronous mode.

In synchronous mode the webservice waits until all request objects are processed then responds. This is suitable for small, fast requests but not for larger requests, due to the fact that a timeout will probably occur if the request takes longer than 1 minute.

In asynchronous mode the webservice responds immediately with a generic 'OK' message and a job ID to be used in future fetch requests. It is then up to the client to make 'fetch' calls to retrieve the results. Each fetch call returns the response objects for those requests that have completed.

For example, consider a client submits a requests object with 100 request objects to a server. The server immediately responds with an "OK, I'm on it" response. The client then waits for a few minutes then submits a requests object to the server with a batchJobID feature value of the Job ID and a batchMode feature value of 'fetch'. The server will return a results response for all the request objects that have completed processing.



6.1.1 Example

Request 1:

```
<?xml version="1.0" encoding="utf-8"?>
<execute xmlns="http://ws.namf09.anzlic.org.au">
<ns1:requests id="895" version="0.3" xmlns:ns1="http://namf09.anzlic.org.au">
        <ns1:authentication>
                <ns1:username>ssmith</ns1:username>
                <ns1:password>secret</ns1:password>
        </ns1:authentication>
        <ns1:features>
                <ns1:feature name="batchMode">
                  <ns1:featureValue>asynchronous</ns1:featureValue>
                </ns1:feature>
      </ns1:features>
        <ns1:request id="895.1" name="validateAddress">
                ...
        </ns1:request>
      <ns1:request id="895.2" name="validateAddress">
                ...
        </ns1:request>
      <ns1:request id="895.3" name="validateAddress">
                ...
        </ns1:request>
</ns1:requests>
</execute>
```

Response 1:

```
<?xml version="1.0" encoding="utf-8"?>
<ns2:executeResponse xmlns="http://namf09.anzlic.org.au"
xmlns:ns2="http://ws.namf09.anzlic.org.au">
        <responses id="895">
                <result completed="false" status="OK" hasErrorsInResponseElements="false">
                        <attributes>
                                <attribute name="batchJobID">
                                   <attributeValue>6859340324</attributeValue>
                                </attribute>
                        </attributes>
                </result>
        </responses>
</ns2:executeResponse>
```

Request 2:

```
<?xml version="1.0" encoding="utf-8"?>
<execute xmlns="http://ws.namf09.anzlic.org.au">
<ns1:requests id="895A" version="0.3" xmlns:ns1="http://namf09.anzlic.org.au">
        <ns1:authentication>
                <ns1:username>ssmith</ns1:username>
                <ns1:password>secret</ns1:password>
        </ns1:authentication>
        <ns1:features>
                <ns1:feature name="batchMode">
                  <ns1:featureValue>fetch</ns1:featureValue>
                </ns1:feature>
                <ns1:feature name="batchJobID">
                  <ns1:featureValue>6859340324</ns1:featureValue>
                </ns1:feature>
```

```
      </ns1:features>
</ns1:requests>
</execute>
```

Response 2:

```
<?xml version="1.0" encoding="utf-8"?>
<ns2:executeResponse xmlns="http://namf09.anzlic.org.au"
xmlns:ns2="http://ws.namf09.anzlic.org.au">
        <responses id="895A">
                <result completed="false" status="OK" hasErrorsInResponseElements="false"/>
          <response id="895.1">
                ...
          </response>
          <response id="895.3">
                ...
          </response>
        </responses>
</ns2:executeResponse>
```

Request 3:

```
<?xml version="1.0" encoding="utf-8"?>
<execute xmlns="http://ws.namf09.anzlic.org.au">
<ns1:requests id="895B" version="0.3" xmlns:ns1="http://namf09.anzlic.org.au">
        <ns1:authentication>
                <ns1:username>ssmith</ns1:username>
                <ns1:password>secret</ns1:password>
        </ns1:authentication>
        <ns1:features>
                <ns1:feature name="batchMode">
                  <ns1:featureValue>fetch</ns1:featureValue>
                </ns1:feature>
                <ns1:feature name="batchJobID">
                  <ns1:featureValue>6859340324</ns1:featureValue>
                </ns1:feature>

      </ns1:features>
</ns1:requests>
</execute>
```

Response 3:

```
<?xml version="1.0" encoding="utf-8"?>
<ns2:executeResponse xmlns="http://namf09.anzlic.org.au"
xmlns:ns2="http://ws.namf09.anzlic.org.au">
        <responses id="895A">
                <result completed="true" status="OK" hasErrorsInResponseElements="false"/>
          <response id="895.2">
                ...
          </response>
        </responses>
</ns2:executeResponse>
```

7 Java AXIS 1 Bug

Java's AXIS 1 contains a bug which makes it incompatible to use with the NAMF webservice.

The CallVersions object within the Response object is not handled very well at all. It defaults to a list of CallVersion objects. It should be a list of CallVersions object; a CallVersions object has within it a list of CallVersion objects.This failing is crucial for the response from the getInformation function to contain all the appropriate data.

It is strongly recommended to use Java AXIS 2 instead. Example clients are available for download from the GovDex website (http://www.govdex.gov.au/).

8 Appendix – Full getInformation Response XML Example

```
<?xml version="1.0" encoding="utf-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
	<S:Body>
		<ns2:executeResponse xmlns="http://namf09.anzlic.org.au" xmlns:ns2="http://ws.namf09.anzlic.org.au">
			<responses id="100">
				<result hasErrorsInResponseElements="false" completed="true" status="OK" />
				<response id="100.1">
					<callVersions name="getFieldNames">
						<callVersion>
							<description>This is the Get Field Names Address Web Service.</description>
							<version>0.1</version>
							<features />
							<attributes />
						</callVersion>
					</callVersions>
					<callVersions name="notifyAddress">
						<callVersion>
							<description>This is the Notify Address Web Service.</description>
							<version>0.1</version>
							<features />
							<attributes>
								<attribute name="notificationType">
									<attributeValue>The notification type.</attributeValue>
									<attribute name="optional">
										<attributeValue>FALSE</attributeValue>
									</attribute>
									<attribute name="occurrence">
										<attributeValue>SINGULAR</attributeValue>
```

```
                                        </attribute>
                                        <attribute name="values">
                                                <attribute name="VALUE">
                                                        <attributeValue>I_INSERT</attributeValue>
                                                </attribute>
                                                <attribute name="VALUE">
                                                        <attributeValue>I_ALIAS</attributeValue>
                                                        <attribute name="requirements">
                                                                <attribute name="ATTRIBUTE">

<attributeValue>aliasAddressDetailPID</attributeValue>

                                                                </attribute>
                                                        </attribute>
                                                </attribute>
                                                <attribute name="VALUE">
                                                        <attributeValue>U_ATTRIB_GEO</attributeValue>
                                                </attribute>
                                                <attribute name="VALUE">
                                                        <attributeValue>U_ATTRIB_ONLY</attributeValue>
                                                </attribute>
                                                <attribute name="VALUE">
                                                        <attributeValue>U_GEO_ONLY</attributeValue>
                                                </attribute>
                                                <attribute name="VALUE">
                                                        <attributeValue>X_COMMENT</attributeValue>
                                                        <attribute name="requirements">
                                                                <attribute name="ATTRIBUTE">
                                                                        <attributeValue>comments</attributeValue>
                                                                </attribute>
                                                        </attribute>
                                                </attribute>
                                                <attribute name="VALUE">
                                                        <attributeValue>D_ROAD_RESERVE</attributeValue>
                                                </attribute>
                                                <attribute name="VALUE">
                                                        <attributeValue>D_NOT_EXISTS</attributeValue>
                                                </attribute>
                                        </attribute>
                                </attribute>
                                <attribute name="certainty">
                                        <attributeValue>The certainty about this notification.</attributeValue>
                                        <attribute name="optional">
                                                <attributeValue>FALSE</attributeValue>
                                        </attribute>
                                        <attribute name="occurrence">
                                                <attributeValue>SINGULAR</attributeValue>
```

```
                                                        </attribute>
                                                        <attribute name="values">
                                                                <attribute name="VALUE">
                                                                        <attributeValue>FULL</attributeValue>
                                                                </attribute>
                                                                <attribute name="VALUE">
                                                                        <attributeValue>PARTIAL</attributeValue>
                                                                </attribute>
                                                        </attribute>
                                                </attribute>
                                                <attribute name="effectiveDate">
                                                        <attributeValue>The date the notification comes into effect.</attributeValue>
                                                        <attribute name="optional">
                                                                <attributeValue>TRUE</attributeValue>
                                                        </attribute>
                                                        <attribute name="occurrence">
                                                                <attributeValue>SINGULAR</attributeValue>
                                                        </attribute>
                                                        <attribute name="default">
                                                                <attributeValue>current date</attributeValue>
                                                        </attribute>
                                                        <attribute name="attributes">
                                                                <attribute name="format">
                                                                        <attributeValue>The format of the specified
date.</attributeValue>
                                                                        <attribute name="optional">
                                                                                <attributeValue>TRUE</attributeValue>
                                                                        </attribute>
                                                                        <attribute name="occurrence">
                                                                                <attributeValue>SINGULAR</attributeValue>
                                                                        </attribute>
                                                                        <attribute name="default">
                                                                                <attributeValue>DD/MM/YYYY</attributeValue>
                                                                        </attribute>
                                                                </attribute>
                                                        </attribute>
                                                </attribute>
                                                <attribute name="aliasAddressDetailPID">
                                                        <attributeValue>An address detail pid in GNAF or PAF. eg. GANSW_71732115.
Mandatory when the notification type is I_ALIAS. An error is returned if not supplied.</attributeValue>
                                                        <attribute name="optional">
                                                                <attributeValue>TRUE</attributeValue>
                                                        </attribute>
                                                        <attribute name="occurrence">
                                                                <attributeValue>SINGULAR</attributeValue>
                                                        </attribute>
```

```
                                                </attribute>
                                                <attribute name="comments">
                                                        <attributeValue>A comment about the supplied address. eg. This address is
difficult to find on the ground without a GPS reference or local knowledge. Mandatory when the notification type is X_COMMENT. An error is
returned if not supplied.</attributeValue>
                                                        <attribute name="optional">
                                                                <attributeValue>TRUE</attributeValue>
                                                        </attribute>
                                                        <attribute name="occurrence">
                                                                <attributeValue>SINGULAR</attributeValue>
                                                        </attribute>
                                                </attribute>
                                                <attribute name="areaOfInterest">
                                                        <attributeValue>The area of interest the address belongs to. Mandatory when
the address is specified as an unstructured address. An error is returned if not supplied. Usually not supplied for structured
addresses.</attributeValue>
                                                        <attribute name="optional">
                                                                <attributeValue>TRUE</attributeValue>
                                                        </attribute>
                                                        <attribute name="occurrence">
                                                                <attributeValue>MULTIPLE</attributeValue>
                                                        </attribute>
                                                        <attribute name="values">
                                                                <attribute name="LOOKUP">
                                                                        <attributeValue>areaOfInterest</attributeValue>
                                                                </attribute>
                                                        </attribute>
                                                </attribute>
                                        </attributes>
                                </callVersion>
                        </callVersions>
                        <callVersions name="getServerInformation">
                                <callVersion>
                                        <description>This is to document features about the NAMF Server that don't relate to any
function.</description>
                                        <version>0.2</version>
                                        <features>
                                                <feature name="batchMode">
                                                        <featureValue>The batch mode of the server.</featureValue>
                                                        <feature name="optional">
                                                                <featureValue>TRUE</featureValue>
                                                        </feature>
                                                        <feature name="occurrence">
                                                                <featureValue>SINGULAR</featureValue>
                                                        </feature>
                                                        <feature name="default">
```

```
                                        <featureValue>SYNCHRONOUS</featureValue>
                                </feature>
                                <feature name="values">
                                        <feature name="VALUE">
                                                <featureValue>SYNCHRONOUS</featureValue>
                                        </feature>
                                </feature>
                        </feature>
                </features>
                <attributes />
        </callVersion>
</callVersions>
<callVersions name="getInformation">
        <callVersion>
                <description>This is the Get Information Address Web Service.</description>
                <version>0.3</version>
                <features>
                        <feature name="version">
                                <featureValue>The version(s) to get information about.</featureValue>
                                <feature name="optional">
                                        <featureValue>TRUE</featureValue>
                                </feature>
                                <feature name="occurrence">
                                        <featureValue>MULTIPLE</featureValue>
                                </feature>
                                <feature name="default">
                                        <featureValue>LATEST</featureValue>
                                </feature>
                                <feature name="values">
                                        <feature name="LOOKUP">
                                                <featureValue>version</featureValue>
                                        </feature>
                                        <feature name="VALUE">
                                                <featureValue>LATEST</featureValue>
                                        </feature>
                                        <feature name="VALUE">
                                                <featureValue>ALL</featureValue>
                                        </feature>
                                </feature>
                        </feature>
                </features>
                <attributes />
        </callVersion>
</callVersions>
<callVersions name="getFieldValues">
        <callVersion>
```

```
                                        <description>This is the Get Field Values Address Web Service.</description>
                                        <version>0.1</version>
                                        <features />
                                        <attributes>
                                                <attribute name="fieldName">
                                                        <attributeValue>The name of the field to lookup.</attributeValue>
                                                        <attribute name="optional">
                                                                <attributeValue>FALSE</attributeValue>
                                                        </attribute>
                                                        <attribute name="occurrence">
                                                                <attributeValue>SINGULAR</attributeValue>
                                                        </attribute>
                                                        <attribute name="values">
                                                                <attribute name="LOOKUP">
                                                                        <attributeValue>Use the getFieldNames function to get valid
values.</attributeValue>
                                                                </attribute>
                                                        </attribute>
                                                </attribute>
                                        </attributes>
                                </callVersion>
                        </callVersions>
                        <callVersions name="getNotifications">
                                <callVersion>
                                        <description>This is the Get Notifications Web Service.</description>
                                        <version>0.3</version>
                                        <features>
                                                <feature name="markAsRead">
                                                        <featureValue>If set to true, any returned results will be marked as being
read by the user and not returned in subsequent result sets.</featureValue>
                                                        <feature name="optional">
                                                                <featureValue>TRUE</featureValue>
                                                        </feature>
                                                        <feature name="occurrence">
                                                                <featureValue>SINGULAR</featureValue>
                                                        </feature>
                                                        <feature name="default">
                                                                <featureValue>false</featureValue>
                                                        </feature>
                                                        <feature name="values">
                                                                <feature name="VALUE">
                                                                        <featureValue>TRUE</featureValue>
                                                                </feature>
                                                                <feature name="VALUE">
                                                                        <featureValue>FALSE</featureValue>
                                                                </feature>
```

```
                                                        </feature>
                                                </feature>
                                                <feature name="maxResults">
                                                        <featureValue>Specifies the maximum number of results to return from a
search. The default value is 50 and the absolute maximum is 1000.</featureValue>
                                                        <feature name="optional">
                                                                <featureValue>TRUE</featureValue>
                                                        </feature>
                                                        <feature name="occurrence">
                                                                <featureValue>SINGULAR</featureValue>
                                                        </feature>
                                                        <feature name="default">
                                                                <featureValue>50</featureValue>
                                                        </feature>
                                                </feature>
                                        </features>
                                        <attributes>
                                                <attribute name="startDate">
                                                        <attributeValue>Filter by the date a notification was received. Ignored if
not specified. Can be used in conjunction with endDate.</attributeValue>
                                                        <attribute name="optional">
                                                                <attributeValue>TRUE</attributeValue>
                                                        </attribute>
                                                        <attribute name="occurrence">
                                                                <attributeValue>SINGULAR</attributeValue>
                                                        </attribute>
                                                        <attribute name="attributes">
                                                                <attribute name="format">
                                                                        <attributeValue>The format of the specified
date.</attributeValue>
                                                                        <attribute name="optional">
                                                                                <attributeValue>TRUE</attributeValue>
                                                                        </attribute>
                                                                        <attribute name="occurrence">
                                                                                <attributeValue>SINGULAR</attributeValue>
                                                                        </attribute>
                                                                        <attribute name="default">
                                                                                <attributeValue>DD/MM/YYYY</attributeValue>
                                                                        </attribute>
                                                                </attribute>
                                                        </attribute>
                                                </attribute>
                                                <attribute name="endDate">
                                                        <attributeValue>Filter by the date a notification was received. Ignored if
not specified. Can be used in conjunction with startDate.</attributeValue>
                                                        <attribute name="optional">
```

```
                        <attributeValue>TRUE</attributeValue>
                    </attribute>
                    <attribute name="occurrence">
                        <attributeValue>SINGULAR</attributeValue>
                    </attribute>
                    <attribute name="attributes">
                        <attribute name="format">
                            <attributeValue>The format of the specified
date.</attributeValue>

                            <attribute name="optional">
                                <attributeValue>TRUE</attributeValue>
                            </attribute>
                            <attribute name="occurrence">
                                <attributeValue>SINGULAR</attributeValue>
                            </attribute>
                            <attribute name="default">
                                <attributeValue>DD/MM/YYYY</attributeValue>
                            </attribute>
                        </attribute>
                    </attribute>
                </attribute>
                <attribute name="contributor">
                    <attributeValue>The name of a Contributor to filter the result set with.
Ignored if not supplied.</attributeValue>

                    <attribute name="optional">
                        <attributeValue>TRUE</attributeValue>
                    </attribute>
                    <attribute name="occurrence">
                        <attributeValue>MULTIPLE</attributeValue>
                    </attribute>
                    <attribute name="values">
                        <attribute name="LOOKUP">
                            <attributeValue>contributor</attributeValue>
                        </attribute>
                        <attribute name="VALUE">
                            <attributeValue>ALL</attributeValue>
                        </attribute>
                        <attribute name="VALUE">
                            <attributeValue>ME</attributeValue>
                        </attribute>
                        <attribute name="VALUE">
                            <attributeValue>MY_ORGANISATION</attributeValue>
                        </attribute>
                    </attribute>
                </attribute>
                <attribute name="notificationType">
```

```
                                                        <attributeValue>The notification type to filter the result set with. Ignored
if not specified.</attributeValue>
                                                        <attribute name="optional">
                                                                <attributeValue>TRUE</attributeValue>
                                                        </attribute>
                                                        <attribute name="occurrence">
                                                                <attributeValue>SINGULAR</attributeValue>
                                                        </attribute>
                                                        <attribute name="values">
                                                                <attribute name="VALUE">
                                                                        <attributeValue>I_INSERT</attributeValue>
                                                                </attribute>
                                                                <attribute name="VALUE">
                                                                        <attributeValue>I_ALIAS</attributeValue>
                                                                        <attribute name="requirements">
                                                                                <attribute name="ATTRIBUTE">

        <attributeValue>aliasAddressDetailPID</attributeValue>
                                                                                </attribute>
                                                                        </attribute>
                                                                </attribute>
                                                                <attribute name="VALUE">
                                                                        <attributeValue>U_ATTRIB_GEO</attributeValue>
                                                                </attribute>
                                                                <attribute name="VALUE">
                                                                        <attributeValue>U_ATTRIB_ONLY</attributeValue>
                                                                </attribute>
                                                                <attribute name="VALUE">
                                                                        <attributeValue>U_GEO_ONLY</attributeValue>
                                                                </attribute>
                                                                <attribute name="VALUE">
                                                                        <attributeValue>X_COMMENT</attributeValue>
                                                                        <attribute name="requirements">
                                                                                <attribute name="ATTRIBUTE">
                                                                                        <attributeValue>comments</attributeValue>
                                                                                </attribute>
                                                                        </attribute>
                                                                </attribute>
                                                                <attribute name="VALUE">
                                                                        <attributeValue>D_ROAD_RESERVE</attributeValue>
                                                                </attribute>
                                                                <attribute name="VALUE">
                                                                        <attributeValue>D_NOT_EXISTS</attributeValue>
                                                                </attribute>
                                                        </attribute>
                                                </attribute>
```

```
                                                        <attribute name="ignoreReadStatus">
                                                                <attributeValue>If true, then results previously viewed can be returned,
otherwise only notifications not previously viewed by you will be in the result set.</attributeValue>
                                                                <attribute name="optional">
                                                                        <attributeValue>TRUE</attributeValue>
                                                                </attribute>
                                                                <attribute name="occurrence">
                                                                        <attributeValue>SINGULAR</attributeValue>
                                                                </attribute>
                                                                <attribute name="default">
                                                                        <attributeValue>false</attributeValue>
                                                                </attribute>
                                                                <attribute name="values">
                                                                        <attribute name="VALUE">
                                                                                <attributeValue>TRUE</attributeValue>
                                                                        </attribute>
                                                                        <attribute name="VALUE">
                                                                                <attributeValue>FALSE</attributeValue>
                                                                        </attribute>
                                                                </attribute>
                                                        </attribute>
                                                </attributes>
                                        </callVersion>
                                </callVersions>
                                <notes>
                                        <note priority="INFO" name="versionsDefault">The versions feature has defaulted to 'LATEST'.</note>
                                        <note priority="INFO" name="getInformationSuccess">The get information request completed
successfully.</note>
                                </notes>
                                <status>OK</status>
                        </response>
                </responses>
        </ns2:executeResponse>
    </S:Body>
</S:Envelope>
```