NICC ND 1023 v1.2.2 (2010-11)

NICC Document

NGN; Number Portability Common Database; Web Services Communication & Common XML Features

Note : This standard was originally intended to allow fulfilment of changes to General Condition 18 which were announced by Ofcom via the November 2007 Statement entitled “Telephone number portability for consumers switching suppliers - Concluding Statement”. This change was subsequently set aside by the Competition Appeal Tribunal (Case 1094/3/3/08), and in the April 2010 Statement entitled “Routing calls to ported telephone numbers”, Ofcom concluded that no changes were justified.

However, Ofcom recognised the benefits that a common numbering database approach could bring both to number portability arrangements and to the conservation of geographic numbers, and further concluded that :

> *We consider that a direct routing solution for interconnected fixed networks using such an approach could become viable if and when next generation core network technology is adopted widely by network operators. While the timescale of such adoption is currently uncertain, we would encourage network operators to consider the benefits of incorporating direct routing capability into their next generation network designs.*

Accordingly, whilst NICC Standards cannot warrant what the precise model of usage of a common numbering database for future NGNs will be, this document provides an indication of what was considered appropriate when the issue was considered by NICC, and hence should be borne in mind by network operators when meeting Ofcom’s request to consider direct routeing when designing their NGNs.

Network Interoperability Consultative Committee,
Ofcom,
2a Southwark Bridge Road,
London,
SE1 9HA.

If you have any comments concerning the accuracy of the contents of this document, please write to:

The Technical Secretary,
Network Interoperability Consultative Committee,
Ofcom,
2a Southwark Bridge Road,
London SE1 9HA.

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to NICC.

Pursuant to the NICC IPR Policy, no investigation, including IPR searches, has been carried out by NICC. No guarantee can be given as to the existence of other IPRs which are, or may be, or may become, essential to the present document.

Foreword

This NICC Document (ND) has been produced by NICC TSG NNA.

Introduction

The present document has been produced by the NICC Naming Numbering and Addressing Working Group. It is a protocol framework document, sometimes referred to as a Stage 3 document in the three stage standards development process. There are separate protocol specifications which define the detail of the protocol information flows and this document should be read in conjunction with those specifications.

1 Scope

The present document is a Stage 3 protocol description for the distribution of Data within the NICC Common Numbering Database described in NICC ND1631 [10]. The use of Web Services to implement the information flows across Reference Points M, D_1 and D_2 of NICC ND1631 [10] are described in the present document.

2 References

For the particular version of a document applicable to this release see ND1610 [16].

2.1 Normative references

NOTE: While any hyperlinks included in this clause were valid at the time of publication NICC cannot guarantee their long term validity.

[1] W3C Recommendation (2 May 2001): "XML Schema Part 2: Datatypes".

NOTE: Available at: http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

[2] IETF RFC 3966: "The tel URI for Telephone Numbers".

NOTE: Available at: http://www.ietf.org/rfc/rfc3966.txt.

[3] IETF RFC 3261: "SIP: Session Initiation Protocol".

NOTE: Available at: http://www.ietf.org/rfc/rfc3261.txt.

[4] WS-I Basic Profile Version 1.0: "Final Material".

NOTE: Available at: http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html.

[5] W3C Note (15 March 2001): "Web Services Description Language (WSDL) 1.1".

NOTE: Available at: http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

[6] OASIS Standard 200401 (March 2004): "Web Services Security: SOAP Message Security 1.0 (WS-Security 2004)".

NOTE: Available at: http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf.

[7] W3C Recommendation (12 February 2002): "XML-Signature Syntax and Processing".

NOTE: Available at: http://www.w3.org/TR/2002/REC-xmldsig-core-20020212/.

[8] ISO 4217: "Codes for the representation of currencies and funds".

[9] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

NOTE: Available at: http://www.ietf.org/rfc/rfc3986.txt.

[10] NICC ND1631 NGN; PSTN/ISDN Service Interconnect; Architecture for usage of Common Numbering Database

[11] NICC ND1024 NGN; Number Portability Common Database; Notification and Data Download Web Service

[12] NICC ND1025 NGN; Number Portability Common Database; Management Web Service

[13] NICC ND1628 Securing Data Flows With IPSec For NGN Interconnects

[14] IETF RFC 1952: "GZIP file format specification version 4.3".

NOTE: Available at: http://www.ietf.org/rfc/rfc1952.txt.

[15] IETF RFC 4346 “The Transport Layer Security (TLS) Protocol Version 1.1”

NOTE: Available at: http://www.ietf.org/rfc/rfc4346.txt

[16] NICC, ND1610, Multi-Service Interconnect of UK Next Generation Networks

3 Definitions and abbreviations

3.1 Definitions

For the purposes of the present document, the following terms and definitions apply:

application: computer program that accesses a Web Service

SOAP: no longer an acronym, protocol used for XML messaging

Web Service: software system designed to support interoperable machine-to-machine interaction over a network

Web Service Provider: entity which provides Web Services interfaces to capabilities offered

Web Service Requester: entity which operates Applications that access Web Services

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ND1631 [11] and the following apply:

ETSI	European Telecommunications Standards Institute
IT	Information Technology
OASIS	Organization for the Advancement of Structured Information Standards
OSA	Open Service Access
RFC	Request For Comment
SIP	Session Initiation Protocol
UDDI	Universal Description Discovery and Integration
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WSDL	Web Service Definition Language
WS-I	Web Services-Interoperability Organization
XML	Extensible Markup Language

4 Use of Web Services technologies

This documents specifies the communication mechanism that will be used by CPs and the CDB to exchange XML requests and responses over the D_1, D_2 and M Reference Points defined in NICC NNA ND1631 [10].

The overall function provided by these information flows is to provide an update mechanism for the Central Database which in turn provides an update mechanism for the Common Database so that the contents held in the Central Database may be propagated in a timely manner to all parts of the Common Database.

4.1 Web Service message content

4.1.1 SOAP

All Web Service messages shall send and accept messages that conform to the SOAP use defined in the WS-I Basic Profile [4], using document/literal binding.

4.1.2 XML

All Web Service messages shall send and accept messages that conform to the XML use defined in the WS-I Basic Profile [4].

4.1.3 HTTP

All Web Service messages shall send and accept messages that conform to the HTTP use defined in the WS-I Basic Profile [4].

In addition to HTTP protocol usage and handling defined in the WS-I Basic Profile [4] the following shall apply and supersede any definition in the WS-I Basic Profile [4].

Where a Web Service Requester anticipates the possibility of a “large” response to a web service request, it shall include the ‘Accept-Encoding’ HTTP header in the request and set it to ‘gzip’ to indicate to the Web Service Provider that it will accept responses that have been encoded using the gzip compression method [14].

Where a Web Service Provider receives a web service request with the ‘Accept-Encoding’ HTTP header set to ‘gzip’ then it may respond to that request by encoding the body of the response using the gzip compression method [14] and, when having done so, inform the Web Service Requester by including the ‘Content-Encoding’ HTTP header and setting it to ‘gzip’. The Web Service Provider shall not encode the body of the response using the gzip compression method [14] unless the Web Service Requester has indicated a willingness to receive such an encoded response in its request. Should there be an indication in the request that the Web Service Requester is willing to receive a gzip encoded response then the decision made by the Web Service Provider as to whether to actually invoke the use of the gzip compression method [14] shall be determined by the length of the uncompressed response that would otherwise be sent. The value used to trigger the use of the gzip compression method [14] shall be configurable in increments of ten kilobytes.

Although the WS-I Basic Profile [4] permits the use of a redirect HTTP status code, changes to the location of Web Services endpoints in regards to critical national infrastructure shall not happen without the need for manual intervention and acceptance. Such changes shall be coordinated “out of channel”, that is managed via an auditable, human dependent process, the mechanics of which are outside the scope of this document. Therefore the Web Service Provider shall not respond to any Web Service request with a HTTP status code that begins with a ‘3’. A Web Service Requester shall not follow any redirect response [HTTP status codes that begin with a ‘3’] it receives for a Web Service request. Should such a response be received then the Web Service Requester shall proceed as it would have had it not received any response.

In addition to those HTTP error status codes (codes that start with a ‘4’) specified in the WS-I Basic Profile [4], the Web Service Provider shall be able to indicate to Web Service Requester that either an incorrect location endpoint has been targeted or that the targeted location endpoint is no longer in use. The Web Service Provider shall use the HTTP status code 404 [Not Found] to indicate to the Web Service Requester that an incorrect location endpoint has been targeted. In place of any redirect responses that would have ordinarily have been sent the Web Service Provider shall

use the HTTP status code 410 [Gone] to indicate to the Web Service Requester that the targeted location endpoint is no longer in use.

4.2 Web Service interface definitions

All Web Service interfaces shall be defined using WSDL 1.1 as defined in the WSDL specification [5] and be conformant to the WSDL use defined in WS-I Basic Profile [4].

4.3 Security for Common Database Interfaces

If a message contains an identifier and/or credentials representing the sender of the message then these shall be provided in a manner prescribed by WS-Security [6].

Integrity of the message content MAY be required by the Web Service Provider. If this is required, then this shall be accomplished using XML Digital Signature [7].

The communication exchange between Web Service Requester and Web Service Provider shall be secured in line with the principles set out in ND1628 Securing Data Flows With IPSec For NGN Interconnects [13].

Where SSL [15] is used to secure the communication exchange then the level of security employed shall be at least equivalent to that obtained using IPsec as specified in ND1628 Securing Data Flows With IPSec For NGN Interconnects [13].

5 Exceptions

Exceptions are defined with three data items.

The first data item is a unique identifier for the message. This allows the receiver of the message to recognize the message easily in a language-neutral manner. Thus applications and people seeing the message do not have to understand the message text to be able to identify the message. This is very useful for customer support as well, since it does not depend on the reader to be able to read the language of the message.

The second data item is the message text, including placeholders (i.e. a "%" sign followed by a decimal integer which provides its number) for additional information. This form is consistent with the form for internationalization of messages used by many technologies (operating systems, programming environments, etc.). Use of this form enables translation of messages to different languages independent of program changes. This is well suited for Web Services messages, as a programming language is not defined. Each placeholder number may appear zero or more times.

The third data item is a list of zero or more strings that represent the content to put in each placeholder defined in the message in the second data item. Each string is substituted for the placeholder with the number giving its position within the list, e.g. the fourth string is substituted for each occurrence of "%4" in the relevant second data item. If there are insufficient strings, then the behaviour is as if a sufficient number of empty strings had been appended to the list. Strings not referred to in the second data item are ignored.

The message texts in later sections are examples and, apart from the maximum placeholder number, do not mandate the text to be used.

5.1 Service exception

When a service is not able to process a request, and retrying the request with the same information will also result in a failure, and the issue is not related to a service policy issue, then the service will issue a fault using the ServiceException fault message. A Service Exception uses the letters 'SVC' at the beginning of the message identifier.

Examples of service exceptions include invalid input, lack of availability of a required resource or a processing error.

5.2 Policy exception

When a service is not able to complete because the request fails to meet a policy criterion, then the service will issue a fault using the PolicyException fault message. To clarify how a Policy Exception differs from a Service Exception, consider that all the input to an operation may be valid as meeting the required input for the operation (thus no Service Exception), but using that input in the execution of the service may result in conditions that require the service not to complete. A Policy Exception uses the letters 'POL' at the beginning of the message identifier.

Examples of policy exceptions include privacy violations, requests not permitted under a governing service agreement or input content not acceptable to the service provider.

6 Exception definitions

The "xsd" namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [1]. The use of the name "xsd" is not semantically significant.

Exceptions are defined with numbers from 0001 to 0999, with numbers 0001 to 0199 reserved for common exceptions, 0200 to 0299 reserved for CDB Notification and Data Download Web Services specification use and 0300 to 0399 reserved for CDB Management Web Services specification use. Numbers from "1000" to "9999" may be used by third parties.

6.1 ServiceException

Faults related to the operation of the service, not including policy related faults, result in the return of a ServiceException message. Service exception messages use the reserved message identifier "SVC".

Element name	Element type	Optional	Description
messageId	xsd:string	No	Message identifier, with prefix SVC
Text	xsd:string	No	Message text
Variables	xsd:string [0..unbounded]	Yes	Variables to substitute into Text string

6.1.1 SVC0001: Service error

Element name	Description
messageId	SVC0001
Text	A service error occurred. Error code is %1
variables	%1 Error code from service - meaningful to support, and may be documented in product documentation

6.1.2 SVC0002: Invalid input value

Element name	Description
messageId	SVC0002
Text	Invalid input value for message part %1
variables	%1 - message part

6.1.3 SVC0003: Invalid input value with list of valid values

Element name	Description
messageId	SVC0003
text	Invalid input value for message part %1, valid values are %2
variables	%1 - message part %2 - list of valid values

6.2 PolicyException

Faults related to policies associated with the service, result in the return of a PolicyException message. Policy exception messages use the reserved message identifier "POL".

Element name	Element type	Optional	Description
messageId	xsd:string	No	Message identifier, with prefix POL
text	xsd:string	No	Message text
variables	xsd:string [0..unbounded]	Yes	Variables to substitute into Text string

6.2.1 POL0001: Policy error

Element name	Description
messageId	POL0001
text	A policy error occurred. Error code is %1
variables	%1 Error code from service - meaningful to support, and may be documented in product documentation

6.2.2 POL0002: Subscription error

Element name	Description
messageId	POL0002
text	This Server is not subscribed for Section %1
variables	%1 - Section verification failed for

7 WSDL usage and style

CDB download and management Web Services definitions:

- shall specify services using document forms as described in clause 7.1.
- shall use namespaces as defined in clause 7.2.
- shall follow the authoring style as defined in clause 7.3.
- shall define messages and interfaces as defined in clause 7.4.
- shall define bindings and services as defined in clause 7.5.

7.1 Service definition and documents

Service definitions are expressed using the facilities of WSDL.

7.1.1 Interface sets

A Web Service definition may contain one or more interfaces (or portTypes in WSDL 1.1 [5]). The characteristics of the Web Service being considered will determine whether one interface or multiple interfaces are appropriate.

The term *Interface Set* will be used to describe the group of interfaces that comprise a Web Service. The *Interface Set* provides a mechanism to group a set of related interfaces using well defined conventions for document and namespace naming.

7.1.2 Preparing for document definition

To provide a consistent use of naming within document sets, and across document sets, a number of conventions are defined that rely on a small amount of preparation to be done before creating the documents.

For each *Interface Set*, a *Base Name* is selected. For each interface within the *Interface Set*, a *Short Name* is selected. These names will be used as part of a common naming convention for the related set of documents (should multiple documents be used to define the service) and for definition naming within the documents. This approach ensures name consistency through Web Service evolution, whether it starts with one interface or multiple interfaces.

Base Names and *Short Names* are always defined using only lower case letters, numbers or underscore characters. They must not start with a non-alphabetic character. An underscore should be used to separate words when the name consists of multiple words. These restrictions apply since these names are used in the construction of file names and URI content.

7.1.3 Documents

There are four document types that can be utilized in a Web Service definition. Each has a specific role, and contributes to the goal of supporting a well organized and useful decomposition of the individual elements of a Web Service definition.

7.1.3.1 Document hierarchy

Where a single *Service* document is not used to define the service, figure 1 below shows the relationship between each document type with the arrows denoting the flow of import definition towards the *Service Binding* document. Where a *Service* document is used the complete service definition is located in one document.

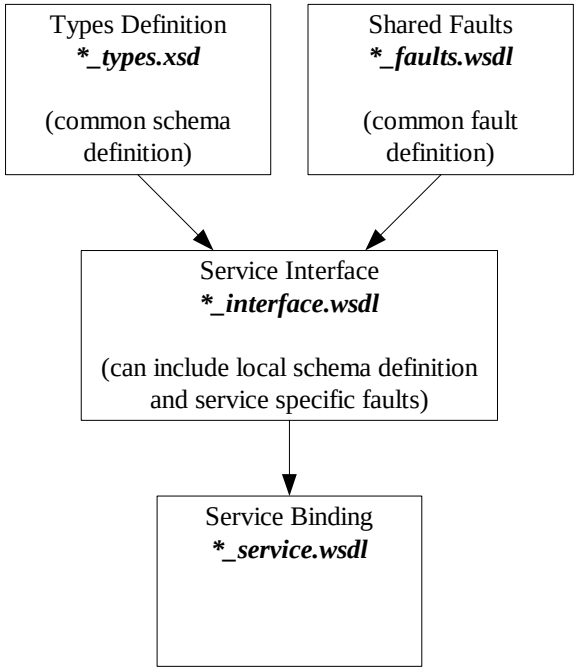

figure 1

7.1.3.2 Types definition document

A *Type Definitions Document* contains data type definitions within a schema namespace.

When the document is related to a specific *Interface Set*, it will use the *Base Name* with the suffix "_types" and the extension ".xsd". When the *Type Definitions Document* is used across multiple *Interface Sets*, it will use an independent name with the suffix "_types" and the extension ".xsd".

7.1.3.3 Shared faults document

A *Shared Faults Document* contains fault definitions that are shared across multiple interfaces in an *Interface Set*, or across *Interface Sets*.

The faults are defined within their own namespace within the WSDL definition namespace. The document name for the present document will use the suffix "_faults" and the extension "wsdl". The first part of the name of the document is based on its usage, with the following guidance:

- If it is used by multiple *Interface Sets*, an independent name reflective of the faults defined will be chosen by the author.
- If it used only by multiple interfaces within an *Interface Set*, then the *Base Name* will be used for the first part of the name.

7.1.3.4 Service interface document

A *Service Interface Document* contains the message and interface (portTypes in WSDL 1.1 [5]) definitions. One interface definition is included in each document. The present document may import *Type Definition Documents* and *Shared Faults Documents*. The present document may be used for a variety of *Service Bindings Documents* without change.

The document name for the present document will use the suffix "_interface" and the extension "wsdl". The name of the document is determined as follows:

- For each interface in an *Interface Set*, the name is a combination of the *Base Name* followed by an underscore followed by the *Short Name* for the interface defined in the present document. Thus multiple documents will have the same *Base Name* as the first portion of the name and the individual interface *Short Name* as the second portion.

7.1.3.5 Service bindings document

A *Service Bindings Document* contains both the binding to be used and the service definition associated with the binding. One service definition is defined in each document. The present document imports one *Service Interface Document*.

The document name for the present document will use the suffix "_service" with the extension "wsdl". Optionally, text representing the specific binding may be added immediately before the "_service" suffix. The name of the document is determined as follows:

- For each interface in an *Interface Set*, the name is a combination of the *Base Name* followed by an underscore followed by the *Short Name* for the interface defined in the present document. Thus multiple documents will have the same *Base Name* as the first portion of the name and the individual interface *Short Name* as the second portion.

7.1.3.6 Service document

A *Service Document* contains the Type, Faults, Interface Sets and Bindings definitions in a single document.

The document name for the present document will be the *BaseName* with the extension “wsdl”.

7.1.4 Document version identifier

It is not always predictable how documents will be stored and used, or when multiple versions of a service may be co-deployed. For this reason, documents may include version identifiers in their naming.

Documents may be assigned a version identifier, corresponding to version information provided in the namespace.

If used, the identifier is added to the end of the name following an underscore. For example, a namespace version of v2_0 would be expressed as _2_0 added to the end of the document name and before its extension.

7.2 Namespaces

The definitions tag has a number of attributes for namespace definitions. These definitions will include a set of common definitions and WSDL specific definitions. The common definitions will be provided in all WSDL documents.

Correct use of namespaces is essential for both creating WSDL that will be usable by a variety of tools, and creating references that allow use of reusable content across the set of documents for a Web Service.

The following are the key namespaces defined:

- XML Schema namespaces for data type definitions.
- Shared fault namespaces, for easy sharing of common fault definitions.
- WSDL interface namespace for Web Service interface definitions.
- WSDL schema local interface namespace for XML Schema definitions contained in the WSDL interface definition.
- WSDL binding namespace for service bindings definitions.

7.2.1 Namespace elements

The namespace definition includes three defined elements - the hierarchical name element, the version element and the namespace type element.

The hierarchical name element provides a fully qualified name in a hierarchical form for the namespace. This element is the Web Service specific information.

If a namespace contains a version number it will be a separate namespace element, immediately following the hierarchical name element, and preceding the namespace type. A version number is based on release numbering, consisting of the lowercase letter "v", followed by a number indicating major version number, followed by an underscore "_", followed by a minor version number. Any numbering beyond the minor version number follows the same convention with an underscore separator. Numbers are not limited to single digits.

Following the version number is the namespace type, which is always the last element in the namespace. The namespace type is one of:

- "faults" for *Shared Faults* documents
- "interface" or "local" for *Service Interface* documents
- "service" for *Service* or *Service Bindings* documents.

The *Type Definitions Document* does not have a namespace type; since it does not share its namespace "schema" (XML Schema definitions in the *Service Interface Document* use the "local" namespace type).

7.2.2 Namespace usage

Maintaining a version number as part of the namespace enables multiple versions of a specification to be identified easily, both by human inspection (reading namespace information) and by machine inspection (parsing namespace).

In addition to the version information being contained in the namespace, the WSDL documents shall also incorporate this same version number in the document file name.

This enables each specification document version to be uniquely identified, ensuring correct composition of documents even when multiple versions are present in a system.

When a specification document, or one of its dependent documents, changes then the version for the specification document is incremented. Incrementing of the major version or minor version number is dependent on the nature of the change (a major version number changes at a release cycle, minor version number changes within a release cycle).

For example, if a specification has a types definition document and an interface definition document, then:

- If the types definitions document is updated, its version will be incremented. Since the interface definition is dependent on the types definition document, its version will be incremented as well.
- If the interface definition document is updated, but there are no changes to the types definition document, then only the interface definition document version is incremented (since the types definition document is not dependent on the interface definition document).

For common documents that are used across multiple specifications, a change in the common document will require updating the specification documents that are dependent on the common document (to update the reference to the common document) and thus their document versions will be incremented as well.

7.2.3 Common namespaces

Each document type has some common namespaces will be used in every instance of that document type.

Type Definition documents

```
xmlns:xsd=http://www.w3.org/2001/XMLSchema
```

Shared Faults documents and *Service Interface* documents

```
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd=http://www.w3.org/2001/XMLSchema
```

Service and *Service Bindings* documents for SOAP

```
xmlns="http://schemas.xmlsoap.org/wsdl/"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xsd=http://www.w3.org/2001/XMLSchema
```

Other bindings will have other namespace definitions that will be common to all *Service* and *Service Bindings* documents using that binding.

7.2.4 Target namespace

The target namespace defines the namespace that is the default namespace for the elements within a document. A special case is the target namespace defined for the schema section within the wsdl:types section of a *Service Interface Document*, where the target namespace applies solely to the XML Schema definitions within this section.

The base namespace for WSDL related elements shall be *http://www.uktel.org.uk/wsdl*. For sub-namespaces, they will extend this namespace. All elements defined within the root namespace are defined within this base target namespace.

The target namespace is the same namespace that will be defined later as the XML Schema or WSDL namespace.

7.2.5 WSDL and Schema namespaces

Namespaces may be defined for the WSDL and Schema elements that are defined within the present document, and for those that are referenced by elements in the present document. For each instance, a pair of namespaces may be defined (if applicable). The WSDL namespace is defined with its *Short Name*. The Schema namespace is defined with the *Short Name* plus the suffix "_xsd".

7.3 Authoring style - Document content and names

7.3.1 General WSDL document information

The following guidelines shall be adhered to when authoring WSDL documents:

- WSDL documents will use UTF-8 as their encoding.
- A date in a comment at the top of the WSDL document will indicate the last revision date of the definition and will adhere to the ISO 8601 Complete Representation Extended Format with UTC indicator (e.g. '2008-06-27T08:29:31Z').

7.3.2 Names

Names should be normal language names, without prefixes (e.g. type or interface markers). The names should be meaningful, and not abbreviated in a way that makes the name hard to understand for users of the WSDL that are not literate in computer programming.

As a guideline, a person using the WSDL should be able to load the WSDL file into an XML viewer and see the names displayed and have reasonable understanding of the content.

This does not preclude the use of commonly understood acronyms within names (e.g. ATM) or commonly used abbreviations (e.g. max). However, the resulting name should still be meaningful.

7.3.3 Case usage for names

Two general cases are provided for, both using mixed case names; one with a leading capital letter, the other with a leading lowercase letter.

Names for all elements (all cases where the text name="Name" is used) will start with a letter and be mixed case, with the leading letter of each word capitalized. Words will not be separated by white space, underscore, hyphen or other non-letter characters.

The following elements will have a leading uppercase letter - simpleType name, complexType name, interface (portType) name, binding name, service name, union element name.

The following elements will have a leading lowercase letter - field names (those names used for elements within other elements), message name (message name portion, service prefix will have uppercase letter if used), message part name, interface operation name, binding operation name.

For example, valid names include "Name", "FirstName", "Name1", "mixedCaseName". Invalid names include "1Name", "NAME", "nAME".

7.3.4 Naming conventions for special names

Some names have special meaning, and are often recognized by a naming convention. For example, in some conventions constants are identified by using all upper case letters and underscores between words.

In WSDL, the case usage for names will be followed as described previously. No other conventions for case usage will be used.

For faults, the fault name will be suffixed with the word "Exception".

7.4 Messages and interfaces (PortTypes)

7.4.1 Messages

Messages are used in the operation elements of interfaces (portTypes), providing the definition of the content that is exchanged on input, output and faults.

Document style Web Services define one input message and one output message, each with one part that references an element defined with XML Schema. These may be combined with fault messages in the definition of operations within an interface.

The XML Schema elements that define the message parts are defined within the wsdl:types section of the *Service Interface Document*. These parts may include references to data types defined in *Type Definition Documents*. Faults specific to the interface defined may also have their messages defined in this manner.

7.4.2 Interfaces (PortTypes)

Interfaces make a set of operations available, and define the messages that will be used for each.

For the request/response message pattern, an interface will have an operation definition that contains a single input message, a single output message and zero or more fault messages, in that order.

7.4.3 Faults (Exceptions)

There are four common types of faults that may be part of interface definitions:

1) SOAP faults that occur before a message is received by the Web Service Provider.
2) Service faults, generated as a result of a system failure, resource failure or rejection of the message (e.g. invalid message content).
3) Policy faults, the result of the Web Service Provider rejecting the request, due to a reason other than those covered by a service fault, and not specific to a service (e.g. privacy).
4) Service specific faults that represent a fault that is not common across services.

In defining interfaces, these faults are represented using the following approach:

- SOAP faults are not defined in the WSDL, their content is defined independently. Usually these faults are generated by intermediaries or as part of the infrastructure (e.g. security subsystem).
- Every operation shall include a ServiceException, providing a common manner in which these faults can be provided back to the requester.
- Every operation shall include a PolicyException, providing a common manner in which these faults can be provided back to the requester.
- Only faults that fall outside the service and policy faults should be provided additional fault definitions - in many cases, no additional fault definitions are required.

7.5 Bindings and service definitions

7.5.1 Binding

The binding defines how the WSDL definitions will be utilized in interacting with the network. The binding defines the protocols and operational style of the binding:

- Transport: in accordance with the WS-I Basic Profile [4], documents shall be exchanged using HTTP ["http://schemas.xmlsoap.org/soap/http"].
- Use: documents shall be exchanged using “literal” representations of elements.
- Style: documents shall be exchanged using the “document” style.

7.5.2 Service definition

Services define an endpoint (port) where the Web Service implementation can be located. The address location defined within the *Service* or *Service Binding* document shall be “*http://localhost/*”. At runtime, the Web Service Requester is expected to determine the address location information through local configuration or through a discovery process, replacing the location information in the default service definition.

Annex A (informative): Examples

An example will demonstrate the naming convention. A group of interfaces for a short messaging service (SMS) are defined. This Web Service contains multiple interfaces.

- An *Interface Set* is defined (SMS Interface Set).
- The *Base Name* for the *Interface Set* is assigned the name "sms".
- Each interface within the *Interface Set* is assigned a *Short Name*:
  - The SendSms interface is assigned the *Short Name* "send".
  - The RetrieveSms interface is assigned the *Short Name* "retrieve".

The example base namespace *www.example.com* is used throughout.

A.1 Document names

The following document set would be produced. Additional assumptions for this example are that there are some data type definitions and that multiple interfaces in the *Interface Set* use a common set of faults.

The names provided include the use of version identifiers, where this Web Service is at the v1_0 level.

- One *Type Definitions Document* - common_types_1_0.xsd.
- One *Shared Faults Document* - common_faults_1_0.wsdl.
- Two *Service Interface Documents* – sms_send_interface_1_0.wsdl and sms_retrieve_interface_1_0.wsdl.
- Two *Service Bindings Documents* – sms_send_service_1_0.wsdl and sms_retrieve_service_1_0.wsdl.
- One *Service Document* – sms_1_0.wsdl.

The two *Service Interface* documents import the *Type Definition* document and *Shared Faults* document. The two *Service Bindings* documents import their respective *Service Interface* document. The one *Service* document includes the complete definition for both the SMS Send and Retrieve services.

A.2 Target namespaces

The following target namespaces would be used within a *Service* or *Service Bindings* document.

- Root namespace: http://www.example.com/wsdl/v1_0/service.
- SMS Service Sub-namespace: http://www.example.com/wsdl/sms/v1_0/service.
- SMS Send multi-level sub-namespace: http://www.example.com/wsdl/sms/send/v1_0/service.
- SMS Retrieve multi-level sub-namespace: http://www.example.com/wsdl/sms/retrieve/v1_0/service.

A.3 WSDL and Schema namespaces

The following Schema namespaces would be referenced within *Service Interface* documents.

- SMS Send:

  ```
  xmlns:sms_xsd="http://www.example.com/schema/sms/send/v1_0"
  ```

- SMS Retrieve:

```
xmlns:retrieve_xsd="http://www.example.com/schema/sms/retrieve/v1_0"
```

A common “SMS” schema could also be created for both *Service Interfaces* documents to import, in place of or in conjunction with the Interface specific schema definitions.

```
xmlns:sms_xsd="http://www.example.com/schema/sms/v1_0"
```

The following WSDL namespaces would be referenced within *Service Binding* documents.

- SMS Send:

  ```
  xmlns:send="http://www.example.com/wsdl/sms/send/v1_0/interface"
  ```

- SMS Retrieve:

  ```
  xmlns:retrieve="http://www.example.com/wsdl/sms/retrieve/v1_0/interface"
  ```

A.4 Local namespaces

The following local namespaces would be used within *Service Interface* documents.

- SMS Send:

  ```
  xmlns:send_local_xsd="http://www.example.com/schema/sms/send/v1_0/local
  ```

- SMS Retrieve:

  ```
  xmlns:retrieve_local_xsd="http://www.example.com/schema/sms/retrieve/v1_0/local
  ```

History

Document history		
<Version>	<Date>	<Milestone>
1.1.1	August 2008	Initial issue
1.2.1	December 2008	Tidy-up and removal of extraneous material following completion of ND1024. Addition of warning text on cover-sheet following decision to set aside regulatory changes
1.2.3	November 2010	Warning regards regulatory status updated