Open Geospatial Consortium Inc.

Date: 2006-11-18

Reference number of this document: OGC 06-095

Version: 0.0.9

Category: OpenGIS® Best Practices Paper

Editor: Ingo Simonis, Johannes Echterhoff

Draft OpenGIS® Web Notification Service Implementation Specification

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.
To obtain additional rights of use, visit http://www.opengeospatial.org/legal/.

Warning

This document is not an OGC Standard. It is distributed for review, experimental implementations and comment. It is subject to change without notice and may not be referred to as an OGC Standard. However, an OGC Best Practices Document is an official position of the OGC membership.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OpenGIS® Implementation Specification
Document subtype:	Best practice paper
Document stage:	Draft proposed version
Document language:	English

License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable

Contents

Page

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

 Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Figures

Page

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Tables

Page

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

i. Preface

This is an OGC Best Practices Document. In the OGC, this document type is a document containing discussion of best practices related to the use and/or implementation of an adopted OGC document or related technology and for release to the public. Best Practices Papers are the official position of the OGC and thus represent an endorsement of the content of the paper.

Suggested additions, changes, and comments on this Best Practices Document are welcome and encouraged. Such suggestions may be submitted by email message or by making suggested changes in an edited copy of this document.

The changes made in this document version, relative to the previous version, are tracked by Microsoft Word, and can be viewed if desired. If you choose to submit suggested changes by editing this document, please first accept all the current changes, and then make your suggested changes with change tracking on.

ii. Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 05-008], which is based on the ISO/IEC Directives, Part 2. Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this specification.

iii. Submitting organizations

The following organizations submitted this document to the Open Geospatial Consortium Inc.

University of Muenster

iv. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Ingo Simonis	University of Muenster
Johannes Echterhoff	University of Muenster

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

v. Revision history

Date	Release	Author	Paragraph modified	Description
05. Aug. 2002	0.0.1	Simonis, Wytzisk		Initial version
19. Aug. 2002	0.0.2	Simonis, Wytzisk	div.	WNS-core schema added, minor changes
21. Aug. 2002	0.0.3	Jeffrey Simon	div.	Minor changes
21. Aug. 2002	0.0.4	Simonis, Wytzisk	7; Appendix	Examples added; message schema added; core schema overdone
22.Aug. 2002	0.0.5	Simonis, Wytzisk	Appendix	Schemata overdone
30. Aug. 2002	0.0.6	Simonis, Wytzisk	6	Registration sequence diagram revised; paragraph status management added
06. Sept. 2002	0.0.7	Simonis, Wytzisk	8.3; Appendix	InstantMessaging as a possible communication channel added; WNS core schema revised
20. Sept. 2002	0.0.8	Simonis, Wytzisk	Appendix	WSDL added, schemata overdone
31. Oct. 2002	0.0.9	Simonis, Wytzisk	6, 7, Appendix	Schemas modified
10. Nov. 2002	0.0.10	Simonis, Wytzisk	div.	Minor changes
02. Jan. 2003	0.0.11 (03-008)	Simonis, Wytzisk	Future work	new
18 Jan 2003	0.1.0 (03-008r1)	Harry Niedzwiadek	Various	The final OWS 1.2 review and edit. Produced OGC 03-008r1. Version level incremented to 0.1.0 to reflect successful implementation under OWS 1.2.
July 2006	1.0.0	Ingo Simonis	several	Completely revised version. OGC 03-008r2 is now deprecated.
Nov. 2006	1.0.0	Johannes Echterhoff	several	Revised operations, inserted diagrams + schema

vi. Changes to the OGC Abstract Specification

The OpenGIS® Abstract Specification does not require changes to accommodate the technical contents of this document.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

vii. Future work

This document contains the specification of the Web Notification Service together with a proposal for a common notification mechanism.

Communication endpoints currently are defined by providing specific parameters or data structures. Future versions of this document might make use of URIs for defining communication endpoints – thus eliminating the need for protocol specific parameters or data structures.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Foreword

This specification was initially developed under the OWS 1.2 initiative. The original version OGC 03-008r2 is now deprecated and replaced by this new version OGC 06-095. This document is part of the SWE (Sensor Web Enablement) suite that consists of the following specifications:

1. **Sensor Model Language (SensorML) –** The general models and XML encodings for sensors.

2. **Observations & Measurements (O&M) -** The general models and XML encodings for sensor observations and measurements

3. **Sensor Observation Service (SOS) –** A service by which a client can obtain observations from one or more sensors/platforms (can be of mixed sensor/platform types). Clients can also obtain information that describes the associated sensors and platforms.

4. **Sensor Planning Service (SPS) –** A service by which a client can determine collection feasibility for a desired set of collection requests for one or more mobile sensors/platforms, or a client may submit collection requests directly to these sensors/platforms.

5. **Sensor Alert Service (SAS) –** A service by which a client subscribe for specific and/or self defined alert conditions and gets notified in case the condition is matched. The SAS uses XMPP as transport protocol. This service was defined under SAS.IE 1.0 and is now available as OGC best practice paper OGC 06-28r3.

6. **Web Notification Service (WNS) –** A service by which a client may conduct asynchronous dialogues (message interchanges) with one or more other services. This service is useful when many collaborating services are required to satisfy a client request, and/or when significant delays are involved is satisfying the request. This service was defined under OWS 1.2 in support of SPS operations. WNS has broad applicability in many such multi-service applications. It is now used in several SWE scenarios.

This document includes four annexes; Annexes one and two are normative, and Annexes three and four are informative.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The OGC shall not be held responsible for identifying any or all such patent rights.

 Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Introduction

Web Service environments provide a suitable method to gather requested information in an appropriate way. Synchronous transport protocols such as HTTP provide the necessary functionalities to post requests and to receive the respective responses. HTTP is a reliable protocol in the way it ensures the packet delivery, in order, and with a definitive acknowledgement for each delivery or failure. In case of a simple Web Map Service, a user will receive visualized geographic information after a negligible amount of time, or the user will receive an exception message. HTTP satisfies the needs for this kind of processing, without the need for further functionality.

As services become more complex, basic request-response mechanisms need to contend with delays/failures. For example, mid-term or long-term (trans-) actions demand functions to support asynchronous communications between a user and the corresponding service, or between two services, respectively. The Web Notification Service fulfils these needs by forwarding incoming messages (based on HTTP) to the recipient using arbitrary protocols such as email, Short Message Service (SMS), Instant Messaging (IM), automated phone calls or faxes. The WNS acts as a transport transducer: It simply changes the protocol between incoming and outgoing messages. It is not an active alerting service such as the SAS, but might be used by SAS in case that the recipient wants to receive the alert on his mobile phone rather then waiting in front of a internet-connected computer until an alert is sent.

In the Sensor Web Enablement suite, at least two services make use of the WNS. The Sensor Planning Service allows users to task sensors or to obtain information about the feasibility of certain sensor data collection requests. The tasking or feasibility study can be a long lasting process. The SPS uses the WNS to forward the result message of the original query to the client. The delivery transport is transparent to the SPS. The SPS sends a HTTP message to the WNS, which forwards this message on the protocol defined by the client.

The other service making use of WNS is the Sensor Alert Service. Though most messages are pushed to the client using XMPP as transport protocol, in some cases the client wants to receive the information on a device that is not connected to the Internet. In such cases, the SAS will send the message to the WNS without knowing which transport protocol will be used for delivery. The WNS again changes from one transport protocol to another and delivers the message to the client.

Thus, the Web Notification Service can be seen as a Message Transportation Service.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

OpenGIS® Web Notification Service Implementation Specification

1 Scope

This OpenGIS© document specifies interfaces for requesting information describing the capabilities of a Web Notification Service, for managing registrations at such a service, for sending messages to registered users and for retrieving cached messages.

It also proposes a common notification mechanism for OGC web services.

2 Compliance

Compliance with this specification shall be checked using all the relevant tests specified in Annex A (normative).

3 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

ISO 19105:2000, *Geographic information — Conformance and Testing*

OGC 05-008, *OpenGIS® Web Services Common Specification*

This OWS Common Specification contains a list of normative references that are also applicable to this Implementation Specification.

OGC 05-114 *OpenGIS®* Web Notification Service

OGC 05-090 SWE Architecture

OGC 05-088 SOS

OGC 05-087 O&M

OGC 05-086 SensorML

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

In addition to this document, this specification includes several normative XML Schema Document files as specified in Annex B.

4 Terms and definitions

For the purposes of this specification, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 05-008] and shall apply. In addition, the following terms and definitions apply.

There are no terms and definitions.

5 Conventions

Most of the abbreviated terms listed in Subclause 5.1 of the OWS Common Implementation Specification [OGC 05-008] apply to this document, plus the following abbreviated terms.

SOS	Sensor Observation Service
SPS	Sensor Planning Service
WNS	Web Notification Service
SAS	Sensor Alert Service
SWE	Sensor Web Enablement
O&M	Observation and Measurement
SensorML	Sensor Model Language
TML	Transducer Markup Language

5.1 XMLSpy notation

Most diagrams that appear in this specification are presented using an XML schema notation defined by the XMLSpy[1] product and described in this subclause. XML schema diagrams are for informative use only though they shall reflect the accompanied UML and schema perfectly.

5.1.1 Element

A named rectangle represents the most basic part of the XML Schema notation. Each represents an XML “Element” token. Each Element symbol can be elaborated with extra information as shown in the examples below.

[1] XML Spy: http://www.altova.com

 Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

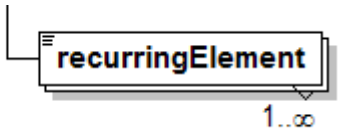
This is a mandatory simple element. Note the upper left corner of the rectangle indicates that data is contained in this element.

5.1.2 Optional Element

Optional (non mandatory) elements are specified with dashed lines used to frame the rectangle.

5.1.3 Recurring Element

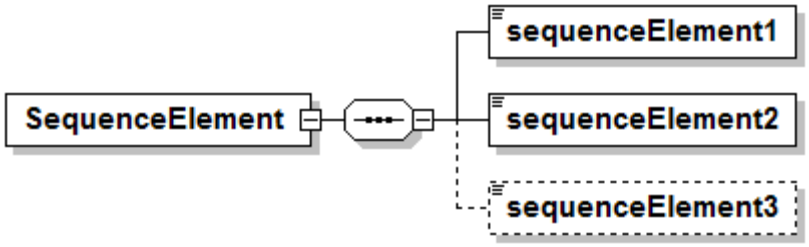
This element (and its child elements if it has any) can occur Multi times.



This example shows a recurring element that must occur at least once but can occur an unlimited amount of times. The upper bound here is shown with the infinity symbol.

5.1.4 Sequence Connector

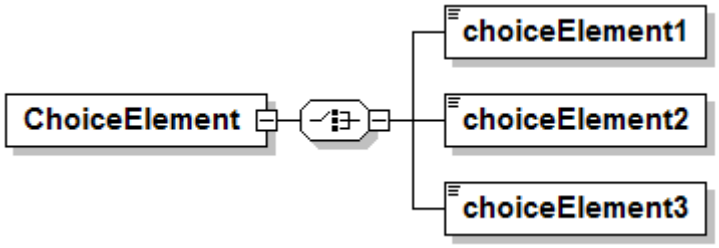
The connection box, called a sequence indicator, indicates that the “SequenceElement” data is made up of three elements. In this example, the first two elements are mandatory and the third element is optional



5.1.5 Choice Connector

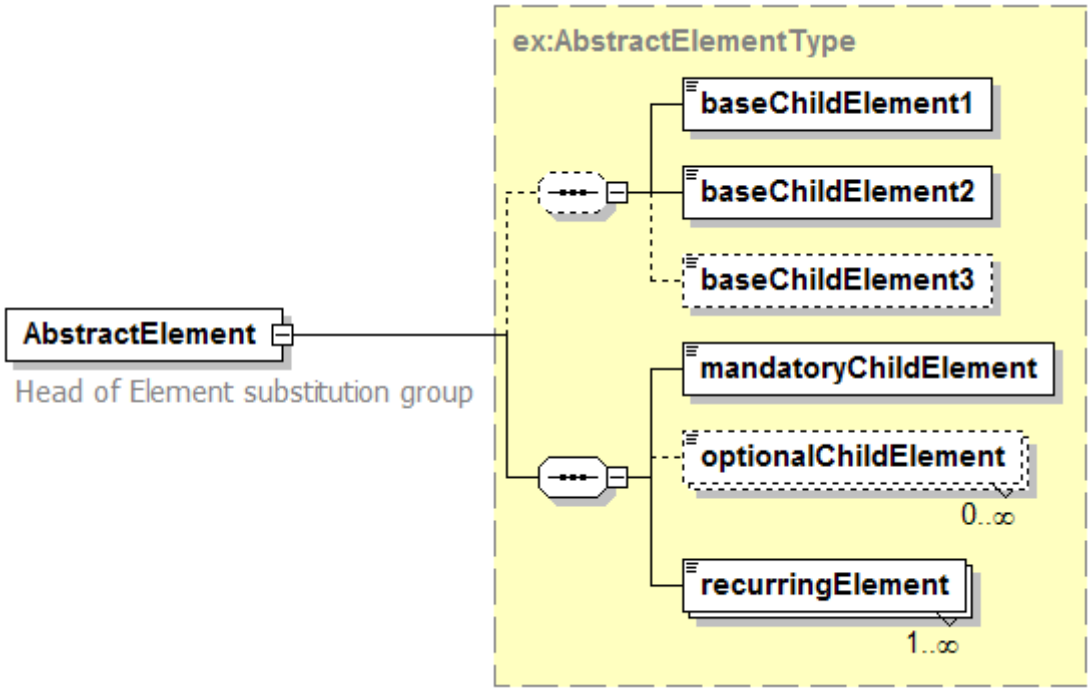
The connection box here is a “choice” indicator, indicating that there is always going to be exactly one of the child elements listed on the right.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.



5.1.6 Definition with Complex Type

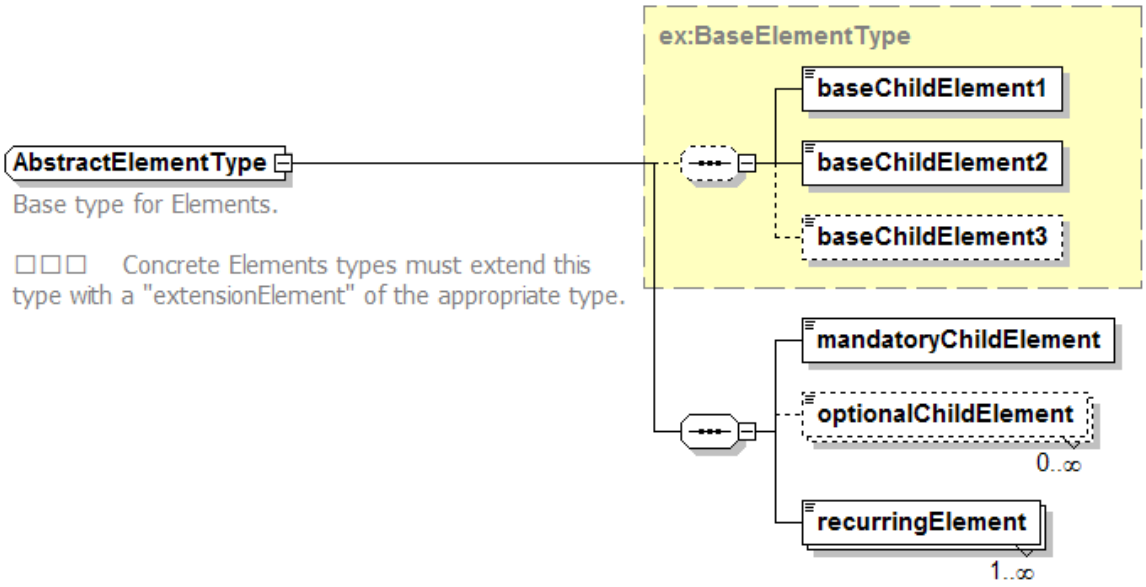
This diagram illustrates the use of a complex type (i.e., “ex:AbstractElementType”) for defining an XML element (e.g., “AbstractElement”).



5.1.7 Complex Type

This diagram illustrates the definition of a complex type (i.e., “AbstractElementType”), extending another complex type (i.e.,“ex:BaseElementType”) with three additional elements. Complex types can be reused to specify that different elements are of the same type.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.



6 WNS overview

The Web Notification Service Model includes two different kinds of communication patterns. First, the “one-way-communication” delivers the message to the client without expecting a response. Second, the “two-way-communication” delivers the message to the client and expects some kind of asynchronous response.

It is important to notice that the WNS handles the message as a black box. The WNS **does not** have any knowledge about the message content.

The basis on which notifications will be sent is free to the service and will be described in its capabilities. The “way-of-notification” palette may include:

- e-mail
- http-call (as HTTP POST: in case of sophisticated clients that act as web services themselves)
- SMS
- XMPP
- phone call
- fax

By default, a WNS provides at least the http transport protocol. The Capabilities document of a WNS announces which additional protocols are supported. To make use of

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

the notification capabilities, users have to be registered beforehand. This registration will be performed by either a user, or by an OGC Service that can act as a proxy for the user, which makes use of the notification functionality (e.g., an SPS). Figure 1 illustrates the two different options. In the first case, an OGC Web service registers a user (this requires that the service has knowledge about the client's message delivery endpoint, e.g. its email address). In the second case, a user/client registers itself directly at the WNS. In both cases, the WNS returns a `registrationID`. This ID, which is unique for every WNS instance, will be used to identify the receiver when a message shall be delivered using the WNS.

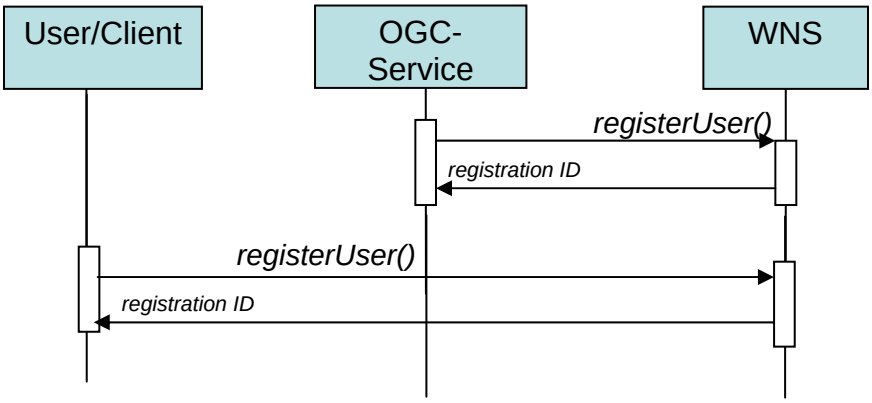


Figure 1 Sequence Diagram of the Registration Process

Independently of the registration requestor (the user or service that acts as a proxy), there is no user data verification mechanism available yet. For example, if a user (client service) requests an expensive data collection, he (the service) cannot get notified if the provided email address was misspelled during registration. In a next evolutionary step, the WNS will be equipped with internal status management and will provide the necessary interface that allows users/services to check if an error has occurred during previous operations.

6.1 Operations

The WNS interface (currently) specifies seven operations that can be requested by a client and performed by a WNS server. These operations are:

a) `GetCapabilities` (mandatory) – This operation allows a client to request and receive back service metadata (or Capabilities) documents that describe the abilities of the specific server implementation. This operation also supports negotiation of the specification version being used for client-server interactions.

b) GetWSDL (optional implementation by servers) – This operation allows a client to request and receive back the WSDL definition of the server interface.

c) `Register` (mandatory) – This operation allows a client to register itself by providing its communication endpoint. We differentiate two cases: a

 Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

`SingleUserRegistration` and a `MultiUserRegistration`. While the former links multiple communication endpoints to a (single) user ID the latter links multiple user IDs to another (multi) user ID, thus creating a group. Any message sent to this group will be delivered to all group members. The WNS is responsible for avoiding circular dependencies between different multi user groups.

d) `Unregister` (mandatory) – This operation allows a client to unregister itself.

e) `UpdateSingleUserRegistration` (optional) – This operation allows a client to update a previous registration by providing a new communication endpoint (e.g. an email address or a telephone number).

f) `UpdateMultiUserRegistration` (optional) – This operation allows a client to update a previous `MultiUserRegistration` by adding or deleting individual group members.

g) `DoNotification` (mandatory) – This operation allows a client to send a message to the WNS, which will be forwarded on the protocol defined by the registered client. In addition to the message, the calling client has to provide the `registrationID` of the registered client.

h) `GetMessage` (mandatory) – This operation allows a client to retrieve a message which has not been delivered by the WNS because of restrictions set by the chosen transport protocol. If notification via SMS or phone call is desired then the WNS will forward the contents of the ShortMessage element of the DoNotification request together with a unique ID assigned to that message (for later retrieval of the complete message via the GetMessage operation).

Each of the WNS operations is described in more detail in subsequent clauses.

6.2 Shared aspects

Many services need a mechanism to perform asynchronous notifications – either as a reply to a previous request from or as an active call to the client. WS-Addressing (GUDGIN et al. 2006) was designed to provide such a mechanism, making it possible to perform a conversation via SOAP messages. Here the default transport protocol is HTTP, but there are also cases when other protocols are used for message delivery, e.g. via e-mail.

But how can OGC web services indicate which protocols they support (even that they support WS-Addressing) and what mechanism can REST services use? This specification contains a separate clause to address these questions. That clause is to be seen separately or in addition to the specification of the WNS operations because it describes schema elements which can be used by any OGC web service which needs to provide a notification mechanism – an underlying WNS implementation is not needed.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

7 WNS operations

7.1 common operation parameters

This clause specifies parameters common for multiple operations.

7.1.1 CommunicationProtocolType

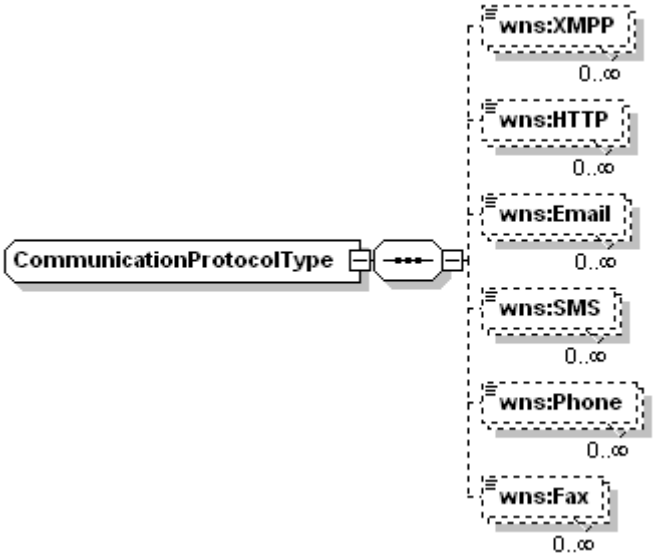


Figure 2: CommunicationProtocolType in XMLSpy notation

The `CommunicationProtocolType` can contain lists of communication endpoints for various protocols. The supported protocols are described in the Capabilities document.

7.1.2 ProtocolsType

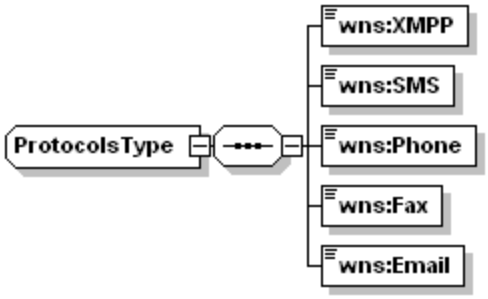


Figure 3: ProtocolsType in XMLSpy notation

The `ProtocolsType` indicates for each possible communication protocol whether it is supported by the WNS or not.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

7.2 Operation request encoding

The encoding of operation requests shall use HTTP GET with KVP encoding and HTTP POST with XML encoding as specified in Clause 11 of [OGC 05-008]. Table 1 summarizes the WNS Service operations and their encoding methods defined in this specification.

Table 1: Operation request encoding

Operation name	Request encoding
GetCapabilities	KVP and XML
GetWSDL	KVP
Register	XML
UpdateSingleUserRegistration	XML
UpdateMultiUserRegistration	XML
DoNotification	XML
GetMessage	KVP and XML

7.3 Operation exceptions

When a WNS server encounters an error while performing one of its operations, it shall return an exception report message as specified in Subclause 8.1 of [OGC 05-008]. The allowed standard exception codes shall include those listed in. For each listed exceptionCode, the contents of the "locator" parameter value shall be as specified in the right column.

NOTE To reduce the need for readers to refer to other documents, the first six values listed below are copied from Table 19 in Subclause 8.3 of [OGC 05-008].

Table 2: WNS exception codes

exceptionCode value	Meaning of code	"locator" value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value [a]	Name of parameter with invalid value
VersionNegotiationFailed	List of versions in "AcceptVersions" parameter value in GetCapabilities operation request did not include any version supported by this server	None, omit "locator" parameter
InvalidUpdateSequence	Value of (optional) updateSequence parameter in GetCapabilities operation request is greater than current value of service metadata updateSequence number	None, omit "locator" parameter
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit "locator" parameter

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

InvalidRequest	Request is not conform to the schema for this operation.	Exception message generated by validator
MessageIDExpired	messageID that has been issued by the client is no longer supported by the service	None, omit "locator" parameter
MessageSendingFailed	An exception occurred while sending a message.	None, omit "locator" parameter
UnknownUserID	One or more of the user (account) IDs provided in the request are not known to the service.	Comma separated list of the unknown IDs.
ProtocolNotSupported	The communication protocol which shall be assigned to a (new or already existing) user account is not supported by the service.	Comma separated list of the protocols not supported ('XMPP', 'SMS', 'Phone', 'Fax' and/or 'Email')
a When an invalid parameter value is received, it seems desirable to place the invalid value(s) in ExceptionText string(s) associated with the InvalidParameterValue value.		

Figure 4 presents which exceptionCode can be applied to which operation.

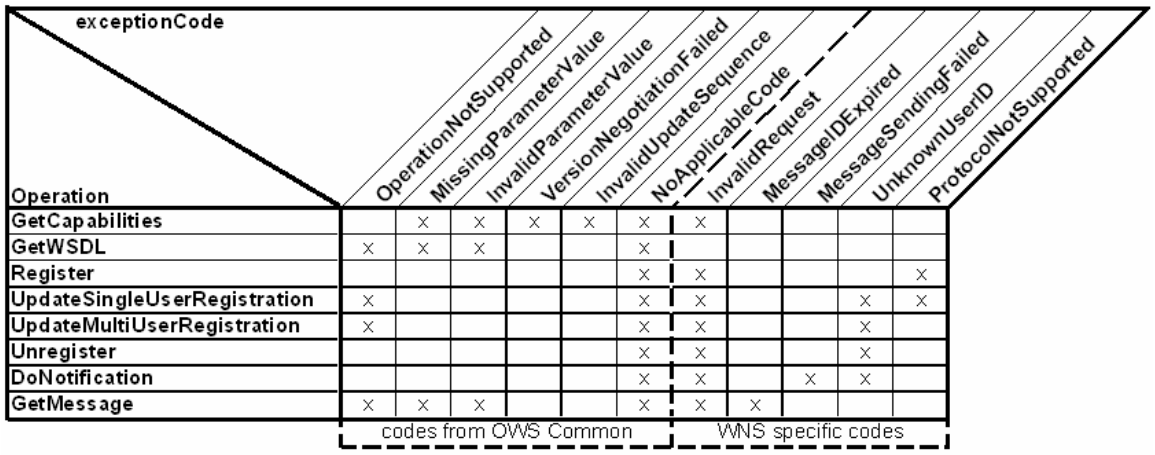


Figure 4: WNS operations with applicable exceptionCodes

7.4 GetCapabilities operation (mandatory)

7.4.1 Introduction

The mandatory GetCapabilities operation allows clients to retrieve service metadata from a server. The response to a GetCapabilities request shall be an XML document containing service metadata about the server, including specific information about WNS. This clause specifies the XML document that a WNS server shall return to describe its capabilities.

 Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

7.4.2 Request

The GetCapabilities operation request shall be as specified in Subclauses 7.2 and 7.3 of [OGC 05-008]. The value of the "service" parameter shall be "WNS". The allowed set of service metadata (or Capabilities) XML document section names and meanings shall be as specified in Tables 3 and 7 of [OGC 05-008], with the addition listed in Table 3 below.

Table 3: Additional Section name values and meanings

Section name	Meaning
Contents	Return contents section in service metadata document

The "Multiplicity and use" column in Table 1 of [OGC 05-008] specifies the optionality of each listed parameter in the GetCapabilities operation request.

All WNS servers shall implement HTTP GET transfer of the GetCapabilities operation request, using KVP encoding. Servers may also implement HTTP POST transfer of the GetCapabilities operation request, using XML encoding only.

7.4.3 Response

The response to a GetCapabilities operation is an XML encoded document. This document provides clients with service metadata about a specific WNS instance.

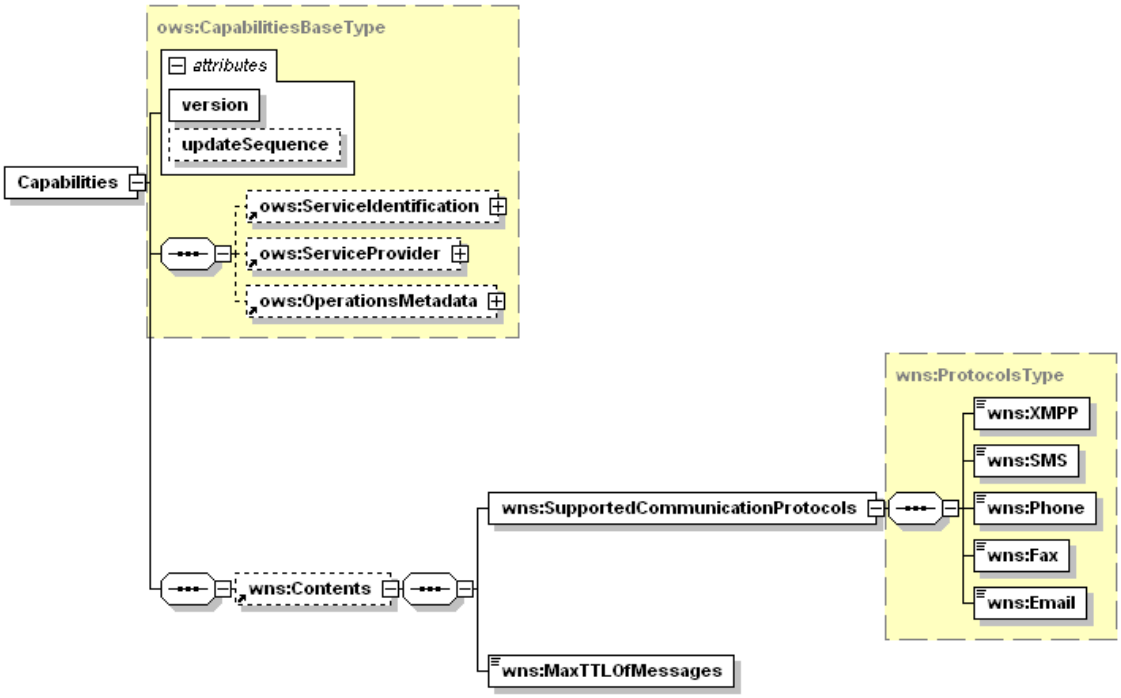


Figure 5: WNS Capabilities document in XMLSpy notation

The portions of the GetCapabilities response document that are defined in the OWS Common specification (see clause 7 in [OGC 05-008]) have not been modified for WNS.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

The Contents section in a WNS Capabilities document shall indicate which communication protocols are supported by the specific service instance and how long messages can be cached if complete delivery was not possible. The parameters of the WNS Contents section are described in the following table.

Table 4: Parameters of the WNS Contents section

Name [a]	Definition	Data type and values	Multiplicity and use
SupportedCommunicationProtocols	The SupportedCommunicationProtocols parameter indicates over which communication protocols the service is able to send messages. HTTP shall be supported by each WNS, the other protocols are optional.	ProtocolsType (see clause 7.1.2)	One (mandatory)
MaxTTLOfMessages	The MaxTTLOfMessages parameter indicates how long messages can be cached if complete delivery was not possible.	XML duration	One (mandatory)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].

7.4.4 Exceptions

Clause 7.3 describes which exceptions are applicable for this operation.

7.4.5 Examples

To request a WNS capabilities document, a client could issue the following KVP encoded GetCapabilities operation request with near-minimum contents:

```
http://www.52north.org:8080/52nWNS/WNS?request=GetCapabilities&service=WNS
```

The corresponding GetCapabilities operation request XML encoded for HTTP POST is:

```
<?xml version="1.0" encoding="UTF-8"?>
<wns:GetCapabilities xmlns:wns="http://www.opengis.net/wns" xmlns:gml="http://www.opengis.net/gml"
xmlns:ows="http://www.opengis.net/ows" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wns
../wnsGetCapabilities.xsd" service="WNS"/>
```

Annex C provides an example of a WNS Capabilities document.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

7.5 GetWSDL operation (optional)

7.5.1 Introduction

The optional GetWSDL operation allows clients to retrieve the WSDL description of the service interface.

7.5.2 Request

Only KVP encoding of the GetWSDL operation request is allowed. Table 5 specifies the parameters of a GetWSDL request.

Table 5: Parameters in the GetWSDL operation request

Name [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty (“WNS”)	One (mandatory)
request	Operation name	Character String type, not empty (“GetWSDL”)	One (mandatory)
version	Specification version for operation	Version of this specification	One (mandatory)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].

7.5.3 Response

The response to a GetWSDL operation request is a WSDL document. This document describes the interface of the service, with the operations supported by the specific service instance.

7.5.4 Exceptions

Clause 7.3 describes which exceptions are applicable for this operation.

7.5.5 Examples

To request the WSDL description of a specific WNS instance, a client could issue the following KVP encoded request:

```
http://www.52north.org:8080/52nWNS/WNS?request=GetWSDL&service=WNS&version=1.0.0
```

Annex D provides an example of a WNS WSDL document.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

7.6 Register operation (mandatory)

7.6.1 Introduction

The Register operation allows WNS clients to either

- register a new user or
- create a group by providing several IDs of registered users.

Basically, each client has to register itself with the WNS in order to receive messages. The registration process simply requires a user name (any string) and the desired communication protocol(s) with their target(s) (e.g. email with developer@52north.org). Further on, the `Register` operation allows grouping of registered users. A message sent to the WNS that contains a `UserID` representing a group will be forwarded to all group members. A group can itself contain other groups – the WNS has to make sure that no circular dependencies occur before a new group account is created.

7.6.2 Request

Figure 6 illustrates the WNS Register operation request.

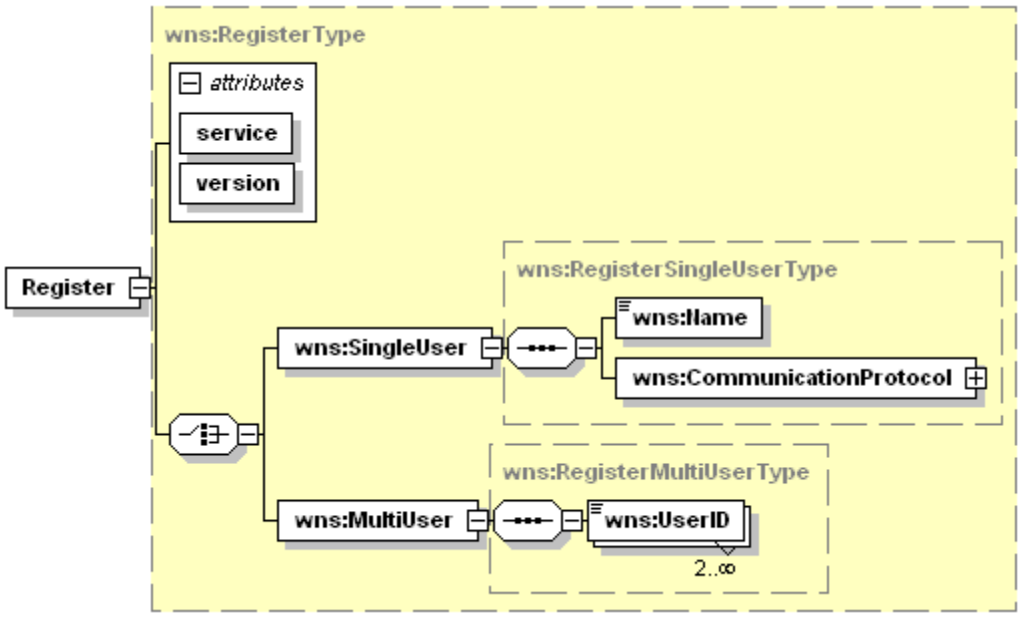


Figure 6: Register request in XMLSpy notation

The parameters of the request are described in the following table.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Table 6: Parameters in the Register operation request

Names [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type: "WNS"	One (mandatory)
version	Specification version for operation	Version of this specification	One (mandatory)
SingleUser	Contains the name of the user and a list of the communication protocols (see clause 7.1.1) desired by the user for message delivery.	RegisterSingleUserType	One (mandatory, if MultiUser is not set)
MultiUser	Contains a list of at least two UserIDs which shall be contained in the new group. All IDs have to belong to users or groups already registered at the service.	RegisterMultiUserType	One (mandatory, if SingleUser is not set)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].			

NOTE The data type of many parameters is specified as "Character String type, not empty". In the XML Schema Documents specified herein, these parameters are encoded with the xsd:string type, which does NOT require that these strings are not empty.

If one or more of the communication protocols / UserIDs contained in the request are not supported / known to the service it shall return an exception with code `ProtocolNotSupported` / `UnknownUserID`.

7.6.3 Response

The response to a Register request contains the UserID assigned by the WNS to the new account.

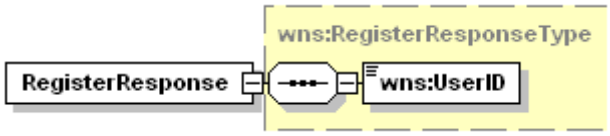


Figure 7: Register operation response in XMLSpy notation

The parameters of the response are described in the following table.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Table 7: Parameters in the Register operation response

Names	Definition	Data type and values	Multiplicity and use
UserID	Unique identifier for a single user or group	Character String type, not empty	One (mandatory)

7.6.4 Exceptions

Clause 7.3 describes which exceptions are applicable for this operation.

7.6.5 Examples

This example request creates a new single user account:

```
<?xml version="1.0" encoding="UTF-8"?>
<Register xmlns="http://www.opengis.net/wns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wns ../wns.xsd" service="WNS" version="1.0.0">
  <SingleUser>
    <Name>Testuser</Name>
    <CommunicationProtocol>
      <Email>testuser@52north.org</Email>
      <Email>testuser@ifgi.uni-muenster.de</Email>
      <SMS>1234567890</SMS>
    </CommunicationProtocol>
  </SingleUser>
</Register>
```

An example response to this request:

```
<?xml version="1.0" encoding="UTF-8"?>
<RegisterResponse xmlns="http://www.opengis.net/wns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/wns ../wns.xsd">
  <UserID>UserID0</UserID>
</RegisterResponse>
```

If one or more of the protocols to register are not supported by the service it would send an `ExceptionReport` like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<ExceptionReport language="en" version="1.0.30" xsi:schemaLocation="http://www.opengis.net/ows
http://schemas.opengeospatial.net/ows/1.0.0/owsExceptionReport.xsd" xmlns="http://www.opengis.net/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <Exception exceptionCode="ProtocolNotSupported" locator="SMS"/>
</ExceptionReport>
```

This example request creates a new multi user account (or group):

```
<?xml version="1.0" encoding="UTF-8"?>
<Register xmlns="http://www.opengis.net/wns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wns ../wns.xsd" service="WNS" version="1.0.0">
  <MultiUser>
    <UserID>UserID1</UserID>
    <UserID>UserID2</UserID>
    <UserID>UserID3</UserID>
  </MultiUser>
</Register>
```


Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

The response could look like the one shown above. Identifiers for single user or group accounts are indistinguishable without further knowledge of the specific service implementation because this specification does not prescribe the syntax of these identifiers.

7.7 Unregister operation (mandatory)

7.7.1 Introduction

The Unregister operation allows WNS clients to unregister a single user or a user group. To Unregister an account, its ID has to be provided.

7.7.2 Request

Figure 8 illustrates the WNS Unregister operation request.

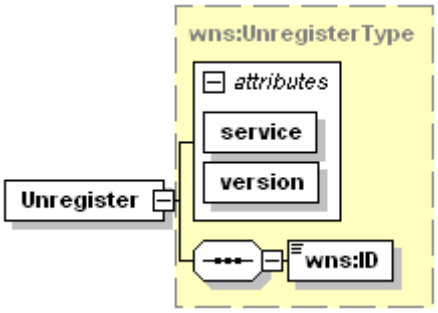


Figure 8: Unregister operation request in XMLSpy notation

The parameters of the request are described in the following table.

Table 8: Parameters in the Unregister operation request

Names [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type: “WNS”	One (mandatory)
version	Specification version for operation	Version of this specification	One (mandatory)
ID	Contains the ID of the account that shall be unregistered.	Character String type, not empty	One (mandatory)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].			

7.7.3 Response

The response to an Unregister request simply indicates whether the operation was a success or could not be fulfilled because the given ID is unknown to the service.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

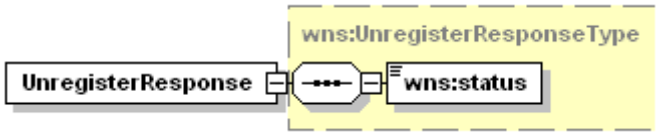


Figure 9: Unregister operation response in XMLSpy notation

The parameters of the response are described in the following table.

Table 9: Parameters in the Unregister operation response

Names	Definition	Data type and values	Multiplicity and use
status	Indicates the outcome of the operation.	Character String type ("success")	One (mandatory)

7.7.4 Exceptions

Clause 7.3 describes which exceptions are applicable for this operation.

7.7.5 Examples

This example request would be used to unregister the account with ID "UserID0":

```
<?xml version="1.0" encoding="UTF-8"?>
<Unregister xmlns="http://www.opengis.net/wns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wns ../wns.xsd" service="WNS" version="1.0.0">
  <ID>UserID0</ID>
</Unregister>
```

If the WNS knows the given ID the response would look like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<UnregisterResponse xmlns="http://www.opengis.net/wns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/wns ../\wns.xsd">
  <status>success</status>
</UnregisterResponse>
```

Otherwise it would send an `ExceptionReport` with exception code `UnknownUserID`:

```
<?xml version="1.0" encoding="UTF-8"?>
<ExceptionReport language="en" version="1.0.30" xsi:schemaLocation="http://www.opengis.net/ows
http://schemas.opengeospatial.net/ows/1.0.0/owsExceptionReport.xsd" xmlns="http://www.opengis.net/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
      <Exception exceptionCode="UnknownUserID" locator="UserID0"/>
</ExceptionReport>
```

7.8 UpdateSingleUserRegistration operation (optional)

7.8.1 Introduction

The UpdateSingleUserRegistration operation allows to update the user settings for an already registered single user (not for a user group - see UpdateMultiUserRegistration for this purpose). Clients may:

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

- update the name and
- add a communication endpoint and
- remove a communication endpoint.

7.8.2 Request

Figure 10 illustrates the request.

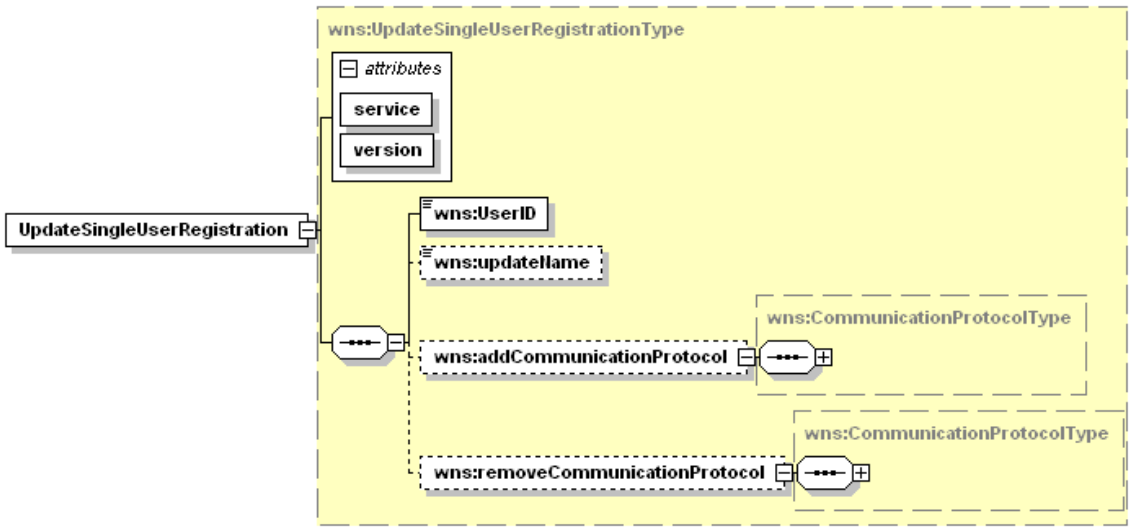


Figure 10: UpdateSingleUserRegistration request in XMLSpy notation

The parameters of the request are described in the following table.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Table 10: Parameters in the UpdateSingleUserRegistration operation request

Names [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type: “WNS”	One (mandatory)
version	Specification version for operation	Version of this specification	One (mandatory)
UserID	ID of the single user account to update.	Character String type, not empty	One (mandatory)
updateName	New name for the account.	Character String type, not empty	One (optional)
addCommunicationProtocol	Contains communication endpoints to add to the account. [b]	CommunicationProtocolType (see clause 7.1.1)	One (optional)
removeCommunicationProtocol	Contains communication endpoints to remove from the account. [b]	CommunicationProtocolType (see clause 7.1.1)	One (optional)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].

b NOTE: if the same communication endpoint appears in the addCommunicationProtocol AND in the removeCommunicationProtocol parameters then the WNS shall remove that endpoint if it already existed and not add it otherwise; i.e., it shall ignore that the endpoint appears in the addCommunicationProtocol parameter.

7.8.3 Response

The normal response to an UpdateSingleUserRegistration request simply indicates that the operation was a success.

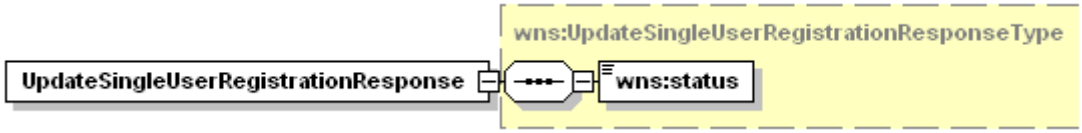


Figure 11: UpdateSingleUserRegistration operation response in XMLSpy notation

The parameters of the response are described in the following table.

Table 11: Parameters in the UpdateSingleUserRegistration operation response

Names	Definition	Data type and values	Multiplicity and use
status	Indicates the outcome of the operation.	Character String type (“success”)	One (mandatory)

All endpoints that are given in the request for removal and that are not existent in the account shall be ignored by the service. In other words, only if communication endpoints are given in the request for being added to an account which belong to protocols not supported by the service, an exception shall be returned (with code

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

`ProtocolNotSupported`) and the request shall be ignored (i.e., the account shall not be updated).

7.8.4 Exceptions

Clause 7.3 describes which exceptions are applicable for this operation.

7.8.5 Examples

An example request for updating the single user account shown in clause 7.6.5 could look like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<UpdateSingleUserRegistration xmlns="http://www.opengis.net/wns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wns
../wns.xsd" service="WNS" version="1.0.0">
  <UserID>UserID0</UserID>
  <updateName>Carl</updateName>
  <addCommunicationProtocol>
    <Email>testuser@52north.org</Email>
    <Email>testuser@gmx.de</Email>
    <SMS>22334455</SMS>
  </addCommunicationProtocol>
  <removeCommunicationProtocol>
    <Email>testuser@52north.org</Email>
    <SMS>1234567890</SMS>
  </removeCommunicationProtocol>
</UpdateSingleUserRegistration>
```

Because the account associated with UserID0 has the communication endpoints testuser@52north.org, testuser@ifgi.uni-muenster.de and the SMS number ‘1234567890’, the request would have the following affects:

- The endpoint testuser@52north.org would be removed (even if it is also contained in the addCommunicationProtocol element - see note contained in **Error! Reference source not found.** for the explanation) as well as the SMS endpoint ‘1234567890’.
- The endpoints testuser@gmx.de and ‘22334455’ would be added to the account.
- The name would be updated to ‘Carl’.

If the WNS supports all the protocols required by the endpoints given in the addCommunicationProtocol element, the response would look like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<UpdateSingleUserRegistrationResponse xmlns="http://www.opengis.net/wns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wns
../\wns.xsd">
  <status>success</status>
</UpdateSingleUserRegistrationResponse>
```

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

7.9 UpdateMultiUserRegistration operation (optional)

7.9.1 Introduction

The `UpdateMultiUserRegistration` operation allows WNS clients to update previously defined MultiUsers, i.e. `UserIDs` that represent more than a single user. Clients are free to add or to remove individual members of the group. The response will be a success message or a list of the unknown `UserIDs` or `MultiUserIDs` respectively.

The WNS has to make sure that no circular dependencies between multi user accounts are introduced with the update.

7.9.2 Request

Figure 12 illustrates the request.

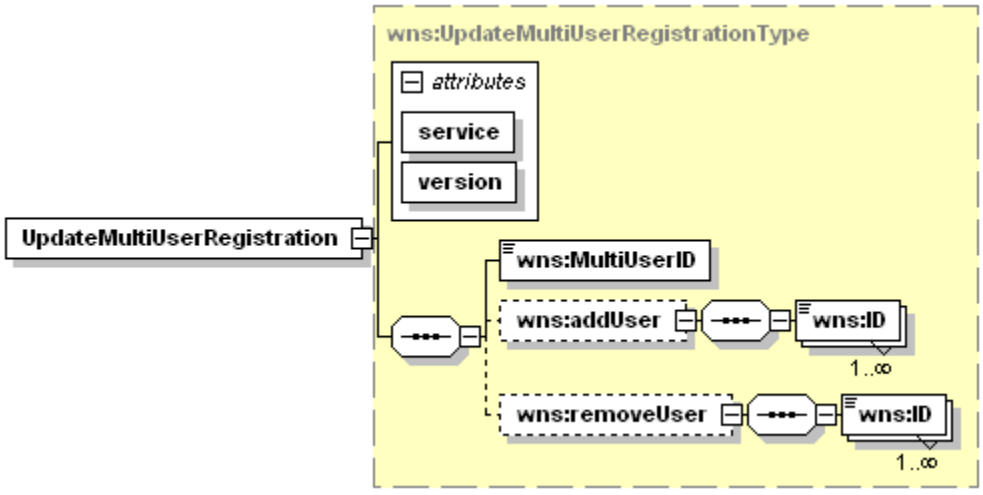


Figure 12: UpdateMultiUserRegistration request in XMLSpy notation

The parameters of the request are described in the following table.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Table 12: Parameters in the UpdateMultiUserRegistration operation request

Names [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type: "WNS"	One (mandatory)
version	Specification version for operation	Version of this specification	One (mandatory)
MultiUserID	ID of the multi user account to update.	Character String type, not empty	One (mandatory)
addUser	List of WNS user account IDs	Character String type, not empty	One to many (optional)
removeUser	List of WNS user account IDs [b]	CommunicationProtocolType (see clause 7.1.1)	One to many (optional)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].

b NOTE: if the same ID appears in the addUser AND in the removeUser parameters then the WNS shall remove that ID if it already existed and not add it otherwise; i.e., it shall ignore that the ID appears in the addUser parameter.

7.9.3 Response

The normal response to an UpdateMultiUserRegistration request simply indicates that the operation was a success.

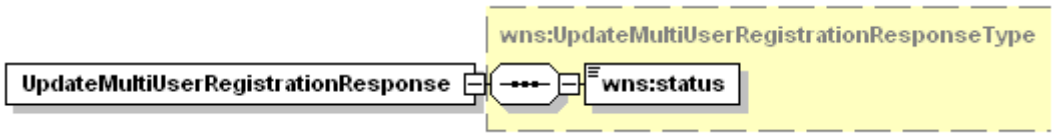


Figure 13: UpdateMultiUserRegistration operation response in XMLSpy notation

The parameters of the response are described in the following table.

Table 13: Parameters in the UpdateMultiUserRegistration operation response

Names	Definition	Data type and values	Multiplicity and use
status	Indicates the outcome of the operation.	Character String type ("success")	One (mandatory)

All user IDs that are given in the request for removal and that are not existent in the multi user account shall be ignored by the service. In other words, only if user IDs are given in the request for being added to a multi user account which are not known to the service, an exception shall be returned (with code `UnknownUserID`) and the request shall be ignored (i.e., the account shall not be updated). This also applies to the ID of the multi user account – if that ID is not known it shall also be listed in the locator element of the exception.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

7.9.4 Exceptions

Clause 7.3 describes which exceptions are applicable for this operation.

7.9.5 Examples

The following request would update the multi user (or group) account with ID 'UserID50':

```
<?xml version="1.0" encoding="UTF-8"?>
<UpdateMultiUserRegistration xmlns="http://www.opengis.net/wns"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wns
../wns.xsd" service="WNS" version="1.0.0">
  <MultiUserID>UserID50</MultiUserID>
  <addUser>
    <ID>UserID0</ID>
    <ID>UserID7</ID>
    <ID>UserID15</ID>
  </addUser>
</UpdateMultiUserRegistration>
```

In case the service does not know the accounts `UserID7` and `UserID15` the response would look like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<ExceptionReport language="en" version="1.0.30" xsi:schemaLocation="http://www.opengis.net/ows
http://schemas.opengeospatial.net/ows/1.0.0/owsExceptionReport.xsd" xmlns="http://www.opengis.net/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <Exception exceptionCode="UnknownUserID" locator="UserID7,UserID15"/>
</ExceptionReport>
```

In this case the account would not have been updated by the service.

7.10 DoNotification operation (mandatory)

7.10.1 Introduction

Messages are being sent to a registered user via the `DoNotification` operation. Some communication protocols are not well suited for transporting XML encoded data. For notification via SMS or phone the WNS will therefore deliver a human readable short message contained in the request and store the complete message for later retrieval by the user via the GetMessage operation (see clause 7.11). For all other protocols, the WNS will deliver the message at once.

7.10.2 Request

Figure 14 illustrates the request.

 Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

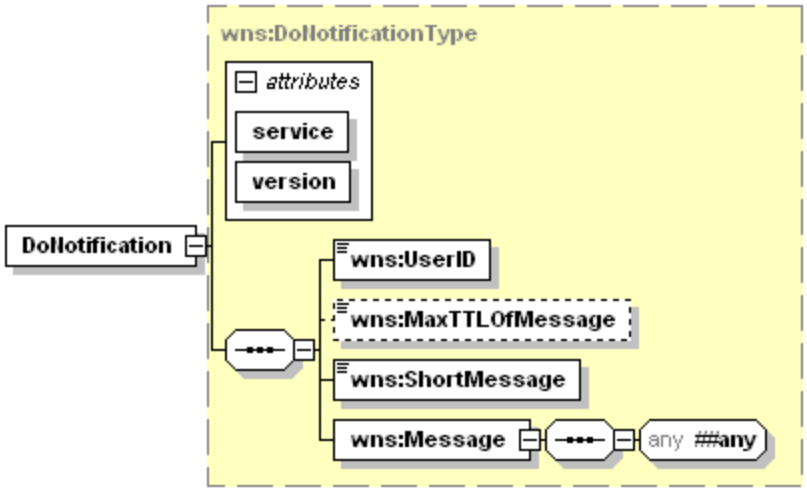


Figure 14: DoNotification request in XMLSpy notation

The parameters of the request are described in the following table.

Table 14: Parameters in the DoNotification operation request

Names [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type: "WNS"	One (mandatory)
version	Specification version for operation	Version of this specification	One (mandatory)
UserID	ID of account where the message shall be delivered to.	Character String type, not empty	One (mandatory)
MaxTTLOfMessage	Indicates for how long the message should be stored by the WNS if immediate delivery is not possible. If that duration is longer than the one the WNS prescribes (see clause **Error! Reference source not found.**), the duration will be reduced accordingly.	XML duration	One (optional)
ShortMessage	Contains a human readable description of the message content. This message will be delivered, together with a MessageID, if SMS or phone shall be used (see clause 7.11)	Character String type, not empty	One (mandatory)
Message	Contains the message which shall be delivered to the user. This XML encoded message can be of any namespace.	Complex (contains arbitrary XML)	One (mandatory)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

7.10.3 Response

The normal response to a DoNotification request simply indicates that the message has been sent successfully. Message delivery might fail because of many reasons that cannot be foreseen by the service – in such a case the service shall send an exception with code `MessageSendingFailed` containing further details.

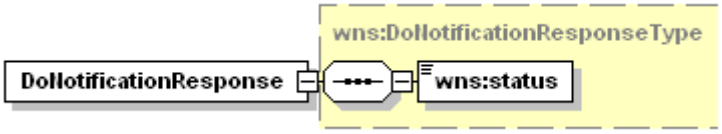


Figure 15: DoNotification operation response in XMLSpy notation

The parameters of the response are described in the following table.

Table 15: Parameters in the DoNotification operation response

Names	Definition	Data type and values	Multiplicity and use
status	Indicates the outcome of the operation.	Character String type (“success”)	One (mandatory)

7.10.4 Exceptions

Clause 7.3 describes which exceptions are applicable for this operation.

7.10.5 Examples

The following DoNotification request contains a message sent by an SPS to ‘UserID0’:

```
<?xml version="1.0" encoding="UTF-8"?>
<DoNotification xmlns="http://www.opengis.net/wns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wns ../wns.xsd" service="WNS" version="1.0.0">
  <UserID>UserID0</UserID>
  <MaxTTLOfMessage>P7D</MaxTTLOfMessage>
  <ShortMessage>Task 1161690976970_684004688690043 is commencing.</ShortMessage>
  <Message>
    <NotificationMessage>
      <ServiceDescription>
        <ServiceType>SPS</ServiceType>
        <ServiceTypeVersion>0.0.30</ServiceTypeVersion>
        <ServiceURL>http://mars.uni-muenster.de:8080/52nSPSv1/SPS</ServiceURL>
      </ServiceDescription>
      <Payload>
        <sps:SPSMessage SPSCorrID="1161690976970_684004688690043"
xmlns:sps="http://www.opengis.net/sps">
          <sps:StatusInformation>
            <sps:status>New data available</sps:status>
          </sps:StatusInformation>
        </sps:SPSMessage>
      </Payload>
    </NotificationMessage>
  </Message>
</DoNotification>
```


Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

In case one of the communication endpoints that the user has registered for is SMS or phone, the message will be stored by the WNS for seven days (unless the service does not allow this) and the short message will be delivered instead. The complete message can later be retrieved by the client via the GetMessage operation (see clause 7.11).

The NotificationMessage element is one of two recommended container elements when sending messages to a client (see clause 0 for further details).

If the WNS was able to deliver the message it will deliver a response like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<DoNotificationResponse xmlns="http://www.opengis.net/wns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/wns ../wns.xsd">
  <status>success</status>
</DoNotificationResponse>
```

7.11 GetMessage operation (optional)

7.11.1 Introduction

A client can retrieve a message stored by the WNS via the `GetMessage` operation. Messages will be stored by the WNS if they could not be delivered completely due to communication protocol restrictions (see clause 7.10). Instead, a short message will be delivered which contains an ID that allows the user to retrieve the message later on.

7.11.2 Request

Figure 16 illustrates the request.

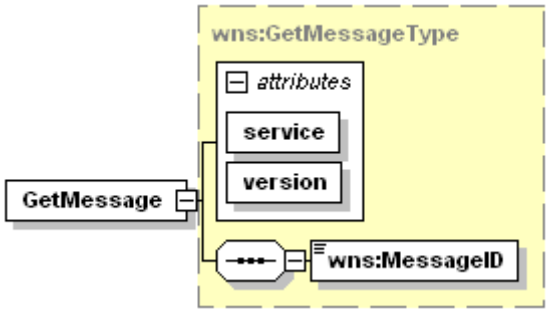


Figure 16: GetMessage request in XMLSpy notation

The parameters of the request are described in the following table.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Table 16: Parameters in the GetMessage operation request

Names [a]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type: “WNS”	One (mandatory)
version	Specification version for operation	Version of this specification	One (mandatory)
MessageID	ID of the message to retrieve.	Character String type, not empty	One (mandatory)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].			

7.11.3 Response

The response simply contains the complete XML encoded message which has been sent in the DoNotification request.

Figure 17: DoNotification operation response in XMLSpy notation

The parameters of the response are described in the following table.

Table 17: Parameters in the DoNotification operation response

Names	Definition	Data type and values	Multiplicity and use
Message	Contains the complete message provided in the DoNotification request.	Complex (contains arbitrary XML)	One (mandatory)

7.11.4 Exceptions

Clause 7.3 describes which exceptions are applicable for this operation.

7.11.5 Examples

The following request would get the message with ID ‘342709’:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetMessage xmlns="http://www.opengis.net/wns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wns ../wns.xsd" service="WNS" version="1.0.0">
  <MessageID>342709</MessageID>
</GetMessage>
```

Same request in KVP encoding:

```
http://www.52north.org:8080/52nWNS/WNS?service=WNS&version=1.0.0&
request=GetMessage&MessageID=342709
```


Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

If the service knows the given ID he will return the message in the response:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetMessageResponse xmlns="http://www.opengis.net/wns" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:schemaLocation="http://www.opengis.net/wns ../wns.xsd">
  <Message>
    <NotificationMessage>
      <ServiceDescription>
        <ServiceType>SPS</ServiceType>
        <ServiceTypeVersion>0.0.30</ServiceTypeVersion>
        <ServiceURL>http://mars.uni-muenster.de:8080/52nSPSv1/SPS</ServiceURL>
      </ServiceDescription>
      <Payload>
        <sps:SPSMessage SPSCorrID="1161690976970_684004688690043"
xmlns:sps="http://www.opengis.net/sps">
          <sps:StatusInformation>
            <sps:status>New data available</sps:status>
          </sps:StatusInformation>
        </sps:SPSMessage>
      </Payload>
    </NotificationMessage>
  </Message>
</GetMessageResponse>
```

Otherwise, he will issue an exception with code `UnkownUserID`.

8 Common notification mechanism

8.1 Introduction

OGC provides many specifications for Web services that either provide data, the ability to gather data or to process data. With algorithms getting more complex and the amount of data getting bigger we are faced with the problem that communication via HTTP (i.e., sending a request and receiving the response) can no longer be said to happen in a synchronous way. We rather have reached the point where we need mechanisms for asynchronous communication between clients and Web services.

Given that we are using SOAP over HTTP we can use WS-Addressing (GUDGIN et al. 2006), which provides a mechanism for asynchronous communication between Web services. The default transport protocol is HTTP, but a client can also provide destinations that require other protocols.

Here we want to provide a mechanism that supports asynchronous communication between OGC Web services even if SOAP is not used. The schema elements provided in this chapter can be taken as building blocks that can be used:

- to indicate which communication protocols a service supports,
- to indicate which message formats a service supports,
- to provide a common way for services to request communication endpoint information from the client and

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

- to provide a common way to encapsulate messages and add the metadata needed to automatically process the message content.

In the following we are going to describe these schema elements.

8.2 Message parameters

First of all, we introduce common message container elements. These elements contain the XML encoded message while providing a minimum of metadata, which enables the receiver to process the message automatically.

We recommend using these message containers whenever exchanging messages between OGC services (with or without a WNS in the middle).

8.2.1 NotificationMessage

The `NotificationMessage` should be used for one-way communication.

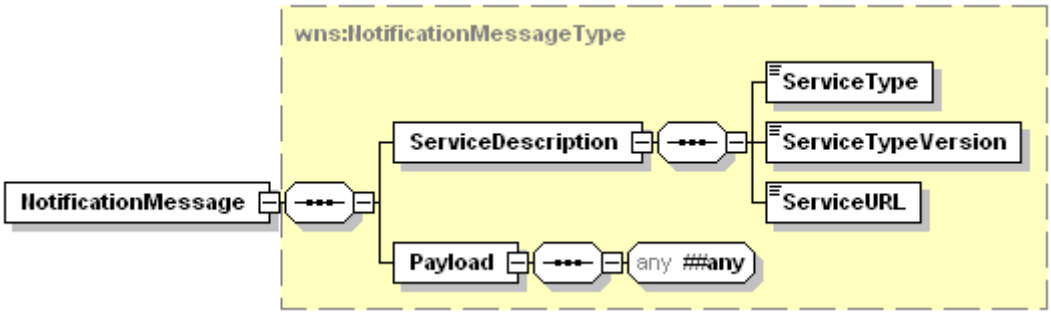


Figure 18: NotificationMessage in XMLSpy notation

The parameters of the response are described in the following table.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Table 18: Parameters in a NotificationMessage

Names[a]	Definition	Data type and values	Multiplicity and use
ServiceDescription	Contains metadata enabling the receiver of the message to automatically process the message. The ServiceType should contain the OGC abbreviation for the service that sent the message, e.g. “SPS” or “SAS”. The ServiceTypeVersion shall indicate the specification version of the sending service (e.g. “1.0.0” or “1.1.0”). The ServiceURL shall contain the URL of the sending service.	Complex: • ServiceType: XML token • ServiceTypeVersion: XML token • ServiceURL: XML anyURI	One (mandatory)
Payload	Contains the actual message. The structure of this message has to be described in the specification of the message originating service.	Any XML element belonging to any namespace.	One (mandatory)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].

8.2.2 CommunicationMessage

The `CommunicationMessage` should be used for two-way-communication. Whenever a reply for a message is needed the sender should make use of the `CommunicationMessage`. The receiver should know what kind of reply message is expected, create the according `ReplyMessage` (see clause 8.2.3) and send it back to the given `CallbackURL`. Whether two-way-communication is supported by a service or not depends on the specification of that service.

Although a WNS can be used to deliver the `CommunicationMessage`, the `ReplyMessage` has to be sent via HTTP Post directly to the given `CallbackURL`.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

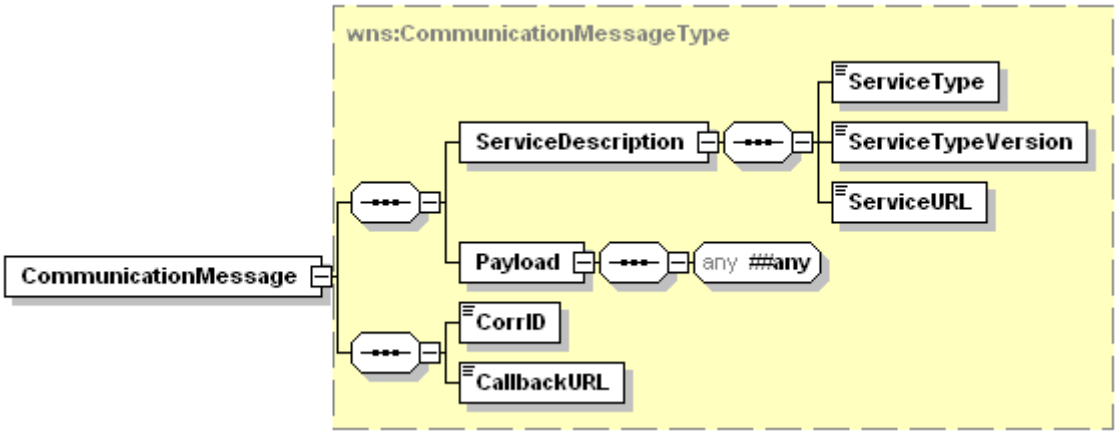


Figure 19: CommunicationMessage in XMLSpy notation

The parameters of the response are described in the following table.

Table 19: Parameters in a CommunicationMessage

Names [a]	Definition	Data type and values	Multiplicity and use
ServiceDescription	Contains metadata enabling the receiver of the message to automatically process the message (see ServiceDescription in Table 18).	Complex (see ServiceDescription in Table 18)	One (mandatory)
Payload	Contains the actual message. The structure of this message has to be described in the specification of the message originating service.	Any XML element belonging to any namespace.	One (mandatory)
CorrID	Contains an ID which has to be included in the reply and which enables the sending service to determine to which communication an incoming reply belongs to.	XML token	One (mandatory)
CallbackURL	Contains the URL where the reply message shall be sent to.	XML anyURI	One (mandatory)

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].

b CorrID and CallbackURL are set by the message originating service

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

8.2.3 ReplyMessage

The `ReplyMessage` is sent by clients as a response upon a `CommunicationMessage`. It contains the correlation ID `CorrID` and the `Payload`. The `Payload` schema and encoding should follow a standardized schema described in the specific service specification.Parameters used by services.

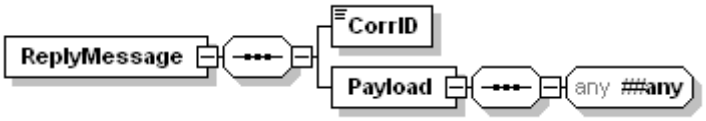


Figure 20: ReplyMessage in XMLSpy notation

The parameters of the response are described in the following table.

Table 20: Parameters in a ReplyMessage

Names [a]	Definition	Data type and values	Multiplicity and use
CorrID	Contains the ID provided in the CommunicationMessage.	XML token	One (mandatory)
Payload	Contains the actual reply message. The structure of this message has to be described in the specification of the message originating service.	Any XML element belonging to any namespace.	One (mandatory)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].			

8.3 Parameters used by services

8.3.1 NotificationAbilities

The `NotificationAbilities` indicate on the one hand which communication protocols a service supports and on the other hand which communication formats he understands. A service can thus include this element in its capabilities so that a client can determine what kind of communication is supported.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

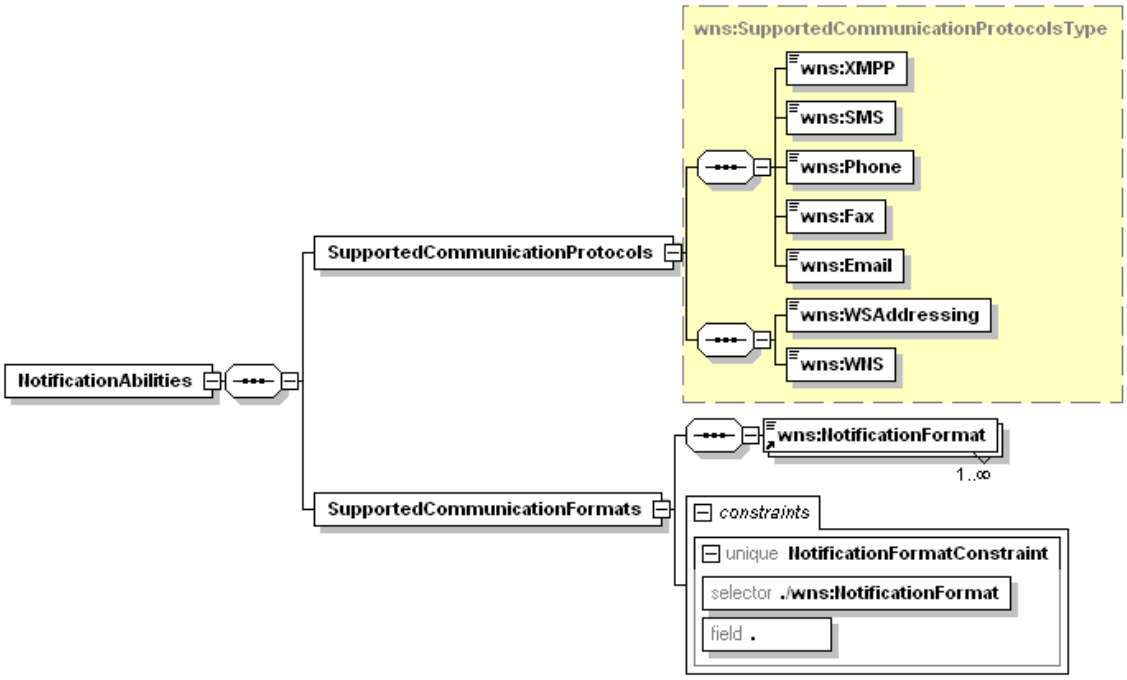


Figure 21: NotificationAbilities in XMLSpy notation

The `SupportedCommunicationProtocols` element should already be known from the WNS Capabilities (see clause **Error! Reference source not found.**). Each element simply indicates whether the specific protocol is supported by the service or not. Here we have two new protocols: `WSAddressing` and `WNS`. The former shows that the service is capable of using WS-Addressing while the latter points out that a client can use his WNS account for being notified by the service.

Having a way to inform a client about the supported communication protocols, the service can also describe which communication formats it supports. Right now the `NotificationTarget` differentiates only between “basic” and “Atom/GeoRSS” message format.

Services that need to send asynchronous messages should specify the structure of those messages. Having a well defined structure also supports automatic consumption of such a message. We call this kind of structure the “basic” message format. The following example shows a simple `NotificationMessage` containing a message from an SPS:

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
<?xml version="1.0" encoding="utf-8"?>
<NotificationMessage xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wns ../wnsShared.xsd" xmlns:sps="http://www.opengis.net/sps"
xmlns="http://www.opengis.net/wns">
  <ServiceDescription>
    <ServiceType>SPS</ServiceType>
    <ServiceTypeVersion>1.0.0</ServiceTypeVersion>
    <ServiceURL>http://www.52north.org/52nSPS/SPS</ServiceURL>
  </ServiceDescription>
  <Payload>
    <sps:SPSMessage>
      <sps:ID>789</sps:ID>
      <sps:Content>
        <sps:StatusInformation>
          <sps:status>New data available</sps:status>
        </sps:StatusInformation>
      </sps:Content>
    </sps:SPSMessage>
  </Payload>
</NotificationMessage>
```

The same message can be incorporated into an “Atom/GeoRSS” feed which can provide further information specifically for human consumption:

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:georss="http://www.georss.org/georss"
xmlns:gml="http://www.opengis.net/gml" xmlns:sps="http://www.opengis.net/sps"
xmlns:wns="http://www.opengis.net/wns" xml:lang="en">
  <title>52N SPS Feed</title>
  <link href="http://www.52north.org:8080/52nSPS/feeds/feed.xml" rel="self"/>
  <link href="http://www.52north.org:8080/52nSPS/SPS" rel="alternate"/>
  <updated>2006-12-13T18:30:02Z</updated>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
  <generator uri="http://www.52north.org:8080/52nSPS/SPS" version="1.0">52N SPS</generator>
  <icon>http://www.52north.org/images/logo.jpg</icon>
  <rights>all</rights>
  <entry>
    <title type="text">Task 789 processed</title>
    <link href="http://www.52north.org:8080/52nSPS/feeds/feed.xml" rel="self"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2006-12-13T18:30:02Z</updated>
    <category term="status"/>
    <content type="html">The data for your task (ID 789) is now available. You can access it via our <a
href="http://www.52north.org/52nSOSClient/SOSClient.jnlp">SOS client</a> or via your standard SPS client.
    </content>
    <georss:where>
      <gml:Envelope>
        <gml:lowerCorner>51.9 6.9</gml:lowerCorner>
        <gml:upperCorner>52.1 7.1</gml:upperCorner>
      </gml:Envelope>
    </georss:where>
    <wns:NotificationMessage>
      <wns:ServiceDescription>
        <wns:ServiceType>SPS</wns:ServiceType>
        <wns:ServiceTypeVersion>1.0.0</wns:ServiceTypeVersion>
        <wns:ServiceURL>http://www.52north.org:8080/52nSPS/SPS</wns:ServiceURL>
      </wns:ServiceDescription>
      <wns:Payload>
        <sps:SPSMessage>
          <sps:ID>789</sps:ID>
          <sps:Content>
            <sps:StatusInformation>
              <sps:status>New data available</sps:status>
            </sps:StatusInformation>
          </sps:Content>
        </sps:SPSMessage>
```

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
      </wns:Payload>
    </wns:NotificationMessage>
  </entry>
</feed>
```

In the future further notification formats may be supported, right now we only have the two formats just presented.

8.3.1 NotificationTarget

The `NotificationTarget` can be included in a service schema whenever there is a need to indicate which communication endpoint(s) can be used for sending messages to.

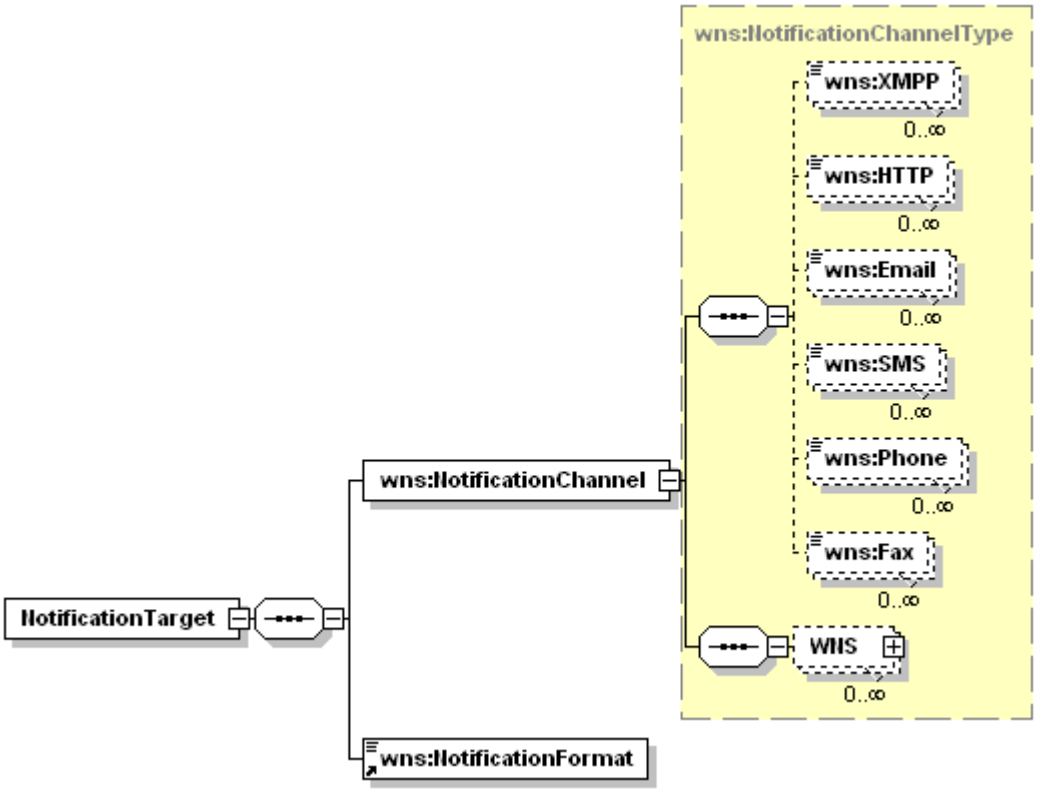


Figure 22: NotificationTarget in XMLSpy notation

Besides the notification channel the `NotificationTarget` also contains an element to indicate which message format is expected.

Note that if a service supports WS-Addressing this would mean that a client can use WS-Addressing instead of using the `NotificationTarget` in the request. That is why service schema should leave the `NotificationTarget` optional and rather specify that it is mandatory if WS-Addressing is used. If it is not optional then WS-Addressing can still be used – the `NotificationChannel` element of the `NotificationTarget` would just be left empty.

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Annex A

(normative)

WNS schema

A.1 wnsAll.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns:wns="http://www.opengis.net/wns" xmlns="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengis.net/wns" elementFormDefault="qualified"
attributeFormDefault="unqualified" xml:lang="en">
 <annotation>
   <appinfo>wnsAll.xsd</appinfo>
   <documentation>
     <description>This XML Schema includes and imports, directly and indirectly, all the XML Schemas defined by
the OGC Web Notification Service (WNS).</description>
   </documentation>
 </annotation>
 <!-- ==================== includes ==================== -->
 <include schemaLocation="wns.xsd"/>
 <include schemaLocation="wnsGetCapabilities.xsd"/>
</schema>
```

A.2 wnsCommon.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:wns="http://www.opengis.net/wns"
targetNamespace="http://www.opengis.net/wns" elementFormDefault="qualified"
attributeFormDefault="unqualified">
 <!-- ==================== TYPES ==================== -->
 <!--Base Type for all operations-->
 <xs:complexType name="BaseOperationType">
   <xs:attribute name="service" type="xs:string" use="required" fixed="WNS"/>
   <xs:attribute name="version" type="xs:string" use="required" fixed="1.0.0"/>
 </xs:complexType>
 <!--CommunicationProtocolType-->
 <xs:complexType name="CommunicationProtocolType">
   <xs:sequence>
     <xs:element name="XMPP" type="xs:token" minOccurs="0" maxOccurs="unbounded"/>
     <xs:element name="HTTP" type="xs:anyURI" minOccurs="0" maxOccurs="unbounded"/>
     <xs:element name="Email" type="xs:anyURI" minOccurs="0" maxOccurs="unbounded"/>
     <xs:element name="SMS" type="xs:unsignedLong" minOccurs="0" maxOccurs="unbounded"/>
     <xs:element name="Phone" type="xs:unsignedLong" minOccurs="0" maxOccurs="unbounded"/>
     <xs:element name="Fax" type="xs:unsignedLong" minOccurs="0" maxOccurs="unbounded"/>
   </xs:sequence>
 </xs:complexType>
 <!--UserIDType-->
 <xs:simpleType name="UserIDType">
   <xs:restriction base="xs:token"/>
 </xs:simpleType>
 <!--ProtocolsType-->
 <xs:complexType name="ProtocolsType">
   <xs:sequence>
     <xs:element name="XMPP" type="xs:boolean"/>
     <xs:element name="SMS" type="xs:boolean"/>
     <xs:element name="Phone" type="xs:boolean"/>
     <xs:element name="Fax" type="xs:boolean"/>
```

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
      <xs:element name="Email" type="xs:boolean"/>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

A.3 wns.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:wns="http://www.opengis.net/wns"
targetNamespace="http://www.opengis.net/wns" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <!-- ==================== Includes and Imports ==================== -->
  <xs:include schemaLocation="./wnsCommon.xsd"/>
  <!-- ==================== R E Q U E S T S ==================== -->
  <!--Register-->
  <xs:element name="Register" type="wns:RegisterType"/>
  <!--UpdateSingleUserRegistration-->
  <xs:element name="UpdateSingleUserRegistration" type="wns:UpdateSingleUserRegistrationType"/>
  <!--UpdateMultiUserRegistration-->
  <xs:element name="UpdateMultiUserRegistration" type="wns:UpdateMultiUserRegistrationType"/>
  <!--Unregister-->
  <xs:element name="Unregister" type="wns:UnregisterType"/>
  <!--DoNotification-->
  <xs:element name="DoNotification" type="wns:DoNotificationType"/>
  <!--GetMessage-->
  <xs:element name="GetMessage" type="wns:GetMessageType"/>
  <!-- ==================== R E S P O N S E S ==================== -->
  <!--RegisterResponse-->
  <xs:element name="RegisterResponse" type="wns:RegisterResponseType"/>
  <!--UpdateSingleUserRegistrationResponse-->
  <xs:element name="UpdateSingleUserRegistrationResponse"
type="wns:UpdateSingleUserRegistrationResponseType"/>
  <!--UpdateMultiUserRegistrationResponse-->
  <xs:element name="UpdateMultiUserRegistrationResponse"
type="wns:UpdateMultiUserRegistrationResponseType"/>
  <!--UnregisterRespone-->
  <xs:element name="UnregisterResponse" type="wns:UnregisterResponseType"/>
  <!--DoNotificationResponse-->
  <xs:element name="DoNotificationResponse" type="wns:DoNotificationResponseType"/>
  <!--GetMessageResponse-->
  <xs:element name="GetMessageResponse" type="wns:GetMessageResponseType"/>
  <!-- ==================== T Y P E S ==================== -->
  <!-- ==================== RequestTypes ==================== -->
  <!--RegisterType-->
  <xs:complexType name="RegisterType">
    <xs:complexContent>
      <xs:extension base="wns:BaseOperationType">
        <xs:choice>
          <xs:element name="SingleUser" type="wns:RegisterSingleUserType"/>
          <xs:element name="MultiUser" type="wns:RegisterMultiUserType"/>
        </xs:choice>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <!--RegisterSingleUserType-->
  <xs:complexType name="RegisterSingleUserType">
    <xs:sequence>
      <xs:element name="Name" type="xs:string"/>
      <xs:element name="CommunicationProtocol" type="wns:CommunicationProtocolType"/>
    </xs:sequence>
  </xs:complexType>
  <!--RegisterMultipleUserType-->
  <xs:complexType name="RegisterMultiUserType">
    <xs:sequence>
      <xs:element name="UserID" type="wns:UserIDType" minOccurs="2" maxOccurs="unbounded"/>
    </xs:sequence>
```


Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
 </xs:complexType>
 <!--UpdateSingleUserRegistrationType-->
 <xs:complexType name="UpdateSingleUserRegistrationType">
   <xs:complexContent>
     <xs:extension base="wns:BaseOperationType">
       <xs:sequence>
         <xs:element name="UserID" type="wns:UserIDType"/>
         <xs:element name="updateName" type="xs:token" minOccurs="0"/>
         <xs:element name="addCommunicationProtocol" type="wns:CommunicationProtocolType"
minOccurs="0"/>
         <xs:element name="removeCommunicationProtocol" type="wns:CommunicationProtocolType"
minOccurs="0"/>
       </xs:sequence>
     </xs:extension>
   </xs:complexContent>
 </xs:complexType>
 <!--UpdateMultiUserRegistrationType-->
 <xs:complexType name="UpdateMultiUserRegistrationType">
   <xs:complexContent>
     <xs:extension base="wns:BaseOperationType">
       <xs:sequence>
         <xs:element name="MultiUserID" type="wns:UserIDType"/>
         <xs:element name="addUser" minOccurs="0">
           <xs:complexType>
             <xs:sequence>
               <xs:element name="ID" type="wns:UserIDType" maxOccurs="unbounded"/>
             </xs:sequence>
           </xs:complexType>
         </xs:element>
         <xs:element name="removeUser" minOccurs="0">
           <xs:complexType>
             <xs:sequence>
               <xs:element name="ID" type="wns:UserIDType" maxOccurs="unbounded"/>
             </xs:sequence>
           </xs:complexType>
         </xs:element>
       </xs:sequence>
     </xs:extension>
   </xs:complexContent>
 </xs:complexType>
 <!--UnregisterType-->
 <xs:complexType name="UnregisterType">
   <xs:complexContent>
     <xs:extension base="wns:BaseOperationType">
       <xs:sequence>
         <xs:element name="ID" type="wns:UserIDType"/>
       </xs:sequence>
     </xs:extension>
   </xs:complexContent>
 </xs:complexType>
 <!--DoNotificationType-->
 <xs:complexType name="DoNotificationType">
   <xs:complexContent>
     <xs:extension base="wns:BaseOperationType">
       <xs:sequence>
         <xs:element name="UserID" type="wns:UserIDType"/>
         <xs:element name="MaxTTLOfMessage" type="xs:duration" minOccurs="0"/>
         <xs:element name="ShortMessage" type="xs:string"/>
         <xs:element name="Message">
           <xs:complexType>
             <xs:sequence>
               <xs:any namespace="##any" processContents="lax"/>
             </xs:sequence>
           </xs:complexType>
         </xs:element>
       </xs:sequence>
     </xs:extension>
```

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
    </xs:complexContent>
  </xs:complexType>
  <!--GetMessageType-->
  <xs:complexType name="GetMessageType">
    <xs:complexContent>
      <xs:extension base="wns:BaseOperationType">
        <xs:sequence>
          <xs:element name="MessageID" type="xs:token">
            <xs:annotation>
              <xs:documentation>Provided by WNS</xs:documentation>
            </xs:annotation>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <!-- ==================== ResponseTypes ==================== -->
  <!--RegisterUserResponseType-->
  <xs:complexType name="RegisterResponseType">
    <xs:sequence>
      <xs:element name="UserID" type="wns:UserIDType">
        <xs:annotation>
          <xs:documentation>unique user id, provided by the WNS</xs:documentation>
        </xs:annotation>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <!--UpdateSingleUserRegistrationResponseType-->
  <xs:complexType name="UpdateSingleUserRegistrationResponseType">
    <xs:sequence>
      <xs:element name="status">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="success"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <!--UpdateMultiUserRegistrationResponseType-->
  <xs:complexType name="UpdateMultiUserRegistrationResponseType">
    <xs:sequence>
      <xs:element name="status">
        <xs:annotation>
          <xs:documentation>unkown MultiUserID if the ID to update is does not exist; unknown UpdateID if one or
more of the IDs to add are unknown</xs:documentation>
        </xs:annotation>
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="success"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
  <!--UnregisterResponseType-->
  <xs:complexType name="UnregisterResponseType">
    <xs:sequence>
      <xs:element name="status">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="success"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
```


Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
 <!--DoNotificationResponseType-->
 <xs:complexType name="DoNotificationResponseType">
   <xs:sequence>
     <xs:element name="status">
       <xs:simpleType>
         <xs:restriction base="xs:token">
           <xs:enumeration value="success"/>
         </xs:restriction>
       </xs:simpleType>
     </xs:element>
   </xs:sequence>
 </xs:complexType>
 <!--GetMessageResponseType-->
 <xs:complexType name="GetMessageResponseType">
   <xs:sequence>
     <xs:element name="Message">
       <xs:complexType>
         <xs:sequence>
           <xs:any namespace="##any" processContents="lax"/>
         </xs:sequence>
       </xs:complexType>
     </xs:element>
   </xs:sequence>
 </xs:complexType>
</xs:schema>
```

A.4 wnsGetCapabilities.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:wns="http://www.opengis.net/wns" xmlns:ows="http://www.opengis.net/ows"
xmlns:xs="http://www.w3.org/2001/XMLSchema" targetNamespace="http://www.opengis.net/wns"
elementFormDefault="qualified" attributeFormDefault="unqualified" xml:lang="en">
 <xs:annotation>
   <xs:appinfo>wnsGetCapabilities.xsd 2005/10/10</xs:appinfo>
   <xs:documentation>
     <description>This XML Schema encodes the WNS GetCapabilities operation request and
response.</description>
   </xs:documentation>
 </xs:annotation>
 <!-- ==================== imports ==================== -->
 <xs:import namespace="http://www.opengis.net/ows"
schemaLocation="http://schemas.opengeospatial.net/ows/1.0.0/owsGetCapabilities.xsd"/>
 <xs:include schemaLocation="./wnsCommon.xsd"/>
 <!-- ==================== elements ==================== -->
 <xs:element name="GetCapabilities">
   <xs:complexType>
     <xs:complexContent>
       <xs:extension base="ows:GetCapabilitiesType">
         <xs:attribute name="service" type="ows:ServiceType" use="required" fixed="WNS"/>
       </xs:extension>
     </xs:complexContent>
   </xs:complexType>
 </xs:element>
 <xs:element name="Capabilities">
   <xs:complexType>
     <xs:complexContent>
       <xs:extension base="ows:CapabilitiesBaseType">
         <xs:sequence>
           <xs:element ref="wns:Contents" minOccurs="0"/>
         </xs:sequence>
       </xs:extension>
     </xs:complexContent>
   </xs:complexType>
 </xs:element>
```

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
  <xs:element name="Contents">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="SupportedCommunicationProtocols" type="wns:ProtocolsType"/>
        <xs:element name="MaxTTLOfMessages" type="xs:duration"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```


Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Annex B

(normative)

Common notification mechanism schema

B.1 wnsShared.xsd

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:wns="http://www.opengis.net/wns" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengis.net/wns" elementFormDefault="qualified"
attributeFormDefault="unqualified">
  <xs:include schemaLocation="./wnsCommon.xsd"/>
  <!-- ==================== ELEMENTS ==================== -->
  <xs:element name="CommunicationMessage" type="wns:CommunicationMessageType"/>
  <xs:element name="NotificationAbilities">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="SupportedCommunicationProtocols"
type="wns:SupportedCommunicationProtocolsType"/>
        <xs:element name="SupportedCommunicationFormats">
          <xs:complexType>
            <xs:sequence>
              <xs:element ref="wns:NotificationFormat" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
          <xs:unique name="NotificationFormatConstraint">
            <xs:selector xpath="./wns:NotificationFormat"/>
            <xs:field xpath="."/>
          </xs:unique>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="NotificationFormat">
    <xs:simpleType>
      <xs:restriction base="xs:string">
        <xs:enumeration value="basic"/>
        <xs:enumeration value="Atom/GeoRSS"/>
      </xs:restriction>
    </xs:simpleType>
  </xs:element>
  <xs:element name="NotificationMessage" type="wns:NotificationMessageType"/>
  <xs:element name="NotificationTarget">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="NotificationChannel" type="wns:NotificationChannelType"/>
        <xs:element ref="wns:NotificationFormat"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="ReplyMessage">
    <xs:annotation>
      <xs:documentation>Sent by the user directly to the callbackURL given in the
CommunicationMessage</xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element name="CorrID" type="xs:token"/>
```

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
        <xs:element name="Payload">
          <xs:complexType>
            <xs:sequence>
              <xs:any namespace="##any" processContents="skip"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <!-- ==================== TYPES ==================== -->
  <xs:complexType name="CommunicationMessageType">
    <xs:complexContent>
      <xs:extension base="wns:WNSMessageType">
        <xs:sequence>
          <xs:element name="CorrID" type="xs:token"/>
          <xs:element name="CallbackURL" type="xs:anyURI"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="NotificationChannelType">
    <xs:complexContent>
      <xs:extension base="wns:CommunicationProtocolType">
        <xs:sequence>
          <xs:element name="WNS" minOccurs="0" maxOccurs="unbounded">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="WNSID" type="xs:token"/>
                <xs:element name="WNSURL" type="xs:anyURI"/>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="NotificationMessageType">
    <xs:complexContent>
      <xs:extension base="wns:WNSMessageType"/>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="SupportedCommunicationProtocolsType">
    <xs:complexContent>
      <xs:extension base="wns:ProtocolsType">
        <xs:sequence>
          <xs:element name="WSAddressing" type="xs:boolean"/>
          <xs:element name="WNS" type="xs:boolean"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="WNSMessageType">
    <xs:sequence>
      <xs:element name="ServiceDescription">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="ServiceType" type="xs:token"/>
            <xs:element name="ServiceTypeVersion" type="xs:token"/>
            <xs:element name="ServiceURL" type="xs:anyURI">
              <xs:annotation>
                <xs:documentation>Helps the user to identify the calling service.</xs:documentation>
              </xs:annotation>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
```


Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
      <xs:element name="Payload">
        <xs:complexType>
          <xs:sequence>
            <xs:any namespace="##any" processContents="skip"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Annex C

(informative)

Example XML documents

C.1 WNS Capabilities example

```
<?xml version="1.0" encoding="UTF-8"?>
<wns:Capabilities xmlns:wns="http://www.opengis.net/wns" xmlns:gml="http://www.opengis.net/gml"
xmlns:ows="http://www.opengis.net/ows" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.opengis.net/wns
../wnsGetCapabilities.xsd" version="1.0.0">
 <ows:ServiceIdentification>
   <ows:Title>IfGI WNS</ows:Title>
   <ows:Abstract>IfGI WNS Server powered by 52 north WNS implementation</ows:Abstract>
   <ows:Keywords>
     <ows:Keyword>message</ows:Keyword>
     <ows:Keyword>protocol changer</ows:Keyword>
     <ows:Keyword>notification</ows:Keyword>
     <ows:Keyword>communication</ows:Keyword>
   </ows:Keywords>
   <ows:ServiceType codeSpace="http://opengeospatial.net">OGC:WNS</ows:ServiceType>
   <ows:ServiceTypeVersion>1.0.0</ows:ServiceTypeVersion>
   <ows:Fees>NONE</ows:Fees>
   <ows:AccessConstraints>NONE</ows:AccessConstraints>
 </ows:ServiceIdentification>
 <ows:ServiceProvider>
   <ows:ProviderName>Institute for Geoinformatics, University of Muenster</ows:ProviderName>
   <ows:ProviderSite xlink:href="http://ifgi.uni-muenster.de"/>
   <ows:ServiceContact>
     <ows:IndividualName>Johannes Echterhoff</ows:IndividualName>
     <ows:PositionName>Research Associate</ows:PositionName>
     <ows:ContactInfo>
       <ows:Phone>
         <ows:Voice>+49 251 83 39761</ows:Voice>
         <ows:Facsimile>+49 251 83 39763</ows:Facsimile>
       </ows:Phone>
       <ows:Address>
         <ows:DeliveryPoint>Institute for Geoinformatics, University of Muenster</ows:DeliveryPoint>
         <ows:City>Muenster</ows:City>
         <ows:AdministrativeArea>NRW</ows:AdministrativeArea>
         <ows:PostalCode>48149</ows:PostalCode>
         <ows:Country>Germany</ows:Country>
         <ows:ElectronicMailAddress>echterhoff@uni-muenster.de</ows:ElectronicMailAddress>
       </ows:Address>
     </ows:ContactInfo>
   </ows:ServiceContact>
 </ows:ServiceProvider>
 <ows:OperationsMetadata>
   <ows:Operation name="GetCapabilities">
     <ows:DCP>
       <ows:HTTP>
         <ows:Get xlink:href="http://server.com/52nWNS/wns?"/>
         <ows:Post xlink:href="http://server.com/52nWNS/wns"/>
       </ows:HTTP>
     </ows:DCP>
     <ows:Parameter name="service">
       <ows:Value>WNS</ows:Value>
```


Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
    </ows:Parameter>
    <ows:Parameter name="version">
      <ows:Value>1.0.0</ows:Value>
    </ows:Parameter>
    <ows:Parameter name="sections">
      <ows:Value>All</ows:Value>
      <ows:Value>ServiceIdentification</ows:Value>
      <ows:Value>ServiceProvider</ows:Value>
      <ows:Value>OperationsMetadata</ows:Value>
      <ows:Value>Contents</ows:Value>
    </ows:Parameter>
  </ows:Operation>
  <ows:Operation name="GetWSDL">
    <ows:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://server.com/52nWNS/wns?"/>
      </ows:HTTP>
    </ows:DCP>
    <ows:Parameter name="service">
      <ows:Value>WNS</ows:Value>
    </ows:Parameter>
    <ows:Parameter name="version">
      <ows:Value>1.0.0</ows:Value>
    </ows:Parameter>
  </ows:Operation>
  <ows:Operation name="Register">
    <ows:DCP>
      <ows:HTTP>
        <ows:Post xlink:href="http://server.com/52nWNS/wns"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="UpdateSingleUserRegistration">
    <ows:DCP>
      <ows:HTTP>
        <ows:Post xlink:href="http://server.com/52nWNS/wns"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="UpdateMultiUserRegistration">
    <ows:DCP>
      <ows:HTTP>
        <ows:Post xlink:href="http://server.com/52nWNS/wns"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="Unregister">
    <ows:DCP>
      <ows:HTTP>
        <ows:Post xlink:href="http://server.com/52nWNS/wns"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="DoNotification">
    <ows:DCP>
      <ows:HTTP>
        <ows:Post xlink:href="http://server.com/52nWNS/wns"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
  <ows:Operation name="GetMessage">
    <ows:DCP>
      <ows:HTTP>
        <ows:Post xlink:href="http://server.com/52nWNS/wns"/>
      </ows:HTTP>
    </ows:DCP>
  </ows:Operation>
```

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
  </ows:OperationsMetadata>
  <wns:Contents>
    <wns:SupportedCommunicationProtocols>
      <wns:XMPP>true</wns:XMPP>
      <wns:SMS>true</wns:SMS>
      <wns:Phone>true</wns:Phone>
      <wns:Fax>true</wns:Fax>
      <wns:Email>true</wns:Email>
    </wns:SupportedCommunicationProtocols>
    <wns:MaxTTLOfMessages>P10D</wns:MaxTTLOfMessages>
  </wns:Contents>
</wns:Capabilities>
```


Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Annex D

(informative)

Example WSDL document

D.1 WNS WSDL document example

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:wns="http://www.opengis.net/wns"
targetNamespace="http://www.opengis.net/wns">
 <types>
   <schema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://schemas.xmlsoap.org/wsdl/" xmlns="http://www.w3.org/2001/XMLSchema">
     <import namespace="http://www.opengis.net/wns" schemaLocation="wnsAll.xsd"/>
   </schema>
 </types>
 <message name="GetCapabilitiesInput">
   <part name="parameter" element="wns:GetCapabilities"/>
 </message>
 <message name="RegisterInput">
   <part name="parameter" element="wns:Register"/>
 </message>
 <message name="UpdateSingleUserRegistrationInput">
   <part name="parameter" element="wns:UpdateSingleUserRegistration"/>
 </message>
 <message name="UpdateMultiUserRegistrationInput">
   <part name="parameter" element="wns:UpdateMultiUserRegistration"/>
 </message>
 <message name="UnregisterInput">
   <part name="parameter" element="wns:Unregister"/>
 </message>
 <message name="DoNotificationInput">
   <part name="parameter" element="wns:DoNotification"/>
 </message>
 <message name="GetMessageInput">
   <part name="parameter" element="wns:GetMessage"/>
 </message>
 <message name="GetCapabilitiesOutput">
   <part name="parameter" element="wns:Capabilities"/>
 </message>
 <message name="RegisterOutput">
   <part name="parameter" element="wns:RegisterResponse"/>
 </message>
 <message name="UpdateSingleUserRegistrationOutput">
   <part name="parameter" element="wns:UpdateSingleUserRegistrationResponse"/>
 </message>
 <message name="UpdateMultiUserRegistrationOutput">
   <part name="parameter" element="wns:UpdateMultiUserRegistrationResponse"/>
 </message>
 <message name="UnregisterOutput">
   <part name="parameter" element="wns:UnregisterResponse"/>
 </message>
 <message name="DoNotificationOutput">
   <part name="parameter" element="wns:DoNotificationResponse"/>
 </message>
```

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
<message name="GetMessageOutput">
  <part name="parameter" element="wns:GetMessageResponse"/>
</message>
<portType name="wnsPortType">
  <operation name="GetCapabilities">
    <input message="wns:GetCapabilitiesInput"/>
    <output message="wns:GetCapabilitiesOutput"/>
  </operation>
  <operation name="Register">
    <input message="wns:RegisterInput"/>
    <output message="wns:RegisterOutput"/>
  </operation>
  <operation name="UpdateSingleUserRegistration">
    <input message="wns:UpdateSingleUserRegistrationInput"/>
    <output message="wns:UpdateSingleUserRegistrationOutput"/>
  </operation>
  <operation name="UpdateMultiUserRegistration">
    <input message="wns:UpdateMultiUserRegistrationInput"/>
    <output message="wns:UpdateMultiUserRegistrationOutput"/>
  </operation>
  <operation name="Unregister">
    <input message="wns:UnregisterInput"/>
    <output message="wns:UnregisterOutput"/>
  </operation>
  <operation name="DoNotification">
    <input message="wns:DoNotificationInput"/>
    <output message="wns:DoNotificationOutput"/>
  </operation>
  <operation name="GetMessage">
    <input message="wns:GetMessageInput"/>
    <output message="wns:GetMessageOutput"/>
  </operation>
</portType>
<binding name="wnsBinding" type="wns:wnsPortType">
  <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
  <operation name="GetCapabilities">
    <soap:operation soapAction="http://wns.foo.bar/SOAP/GetCapabilities"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="Register">
    <soap:operation soapAction="http://wns.foo.bar/SOAP/Register"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="UpdateSingleUserRegistration">
    <soap:operation soapAction="http://wns.foo.bar/SOAP/UpdateSingleUserRegistration"/>
    <input>
      <soap:body use="literal"/>
    </input>
    <output>
      <soap:body use="literal"/>
    </output>
  </operation>
  <operation name="UpdateMultiUserRegistration">
    <soap:operation soapAction="http://wns.foo.bar/SOAP/UpdateSingleUserRegistration"/>
    <input>
      <soap:body use="literal"/>
    </input>
```


Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

```
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="Unregister">
      <soap:operation soapAction="http://wns.foo.bar/SOAP/Unregister"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="DoNotification">
      <soap:operation soapAction="http://wns.foo.bar/SOAP/DoNotification"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
    <operation name="GetMessage">
      <soap:operation soapAction="http://wns.foo.bar/SOAP/GetMessage"/>
      <input>
        <soap:body use="literal"/>
      </input>
      <output>
        <soap:body use="literal"/>
      </output>
    </operation>
  </binding>
  <service name="wnsService">
    <port name="wnsSoap" binding="wns:wnsBinding">
      <soap:address location="http://wns.foo.bar/"/>
    </port>
  </service>
</definitions>
```

Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.

Bibliography

[1] GUDGIN, M., M. HADLEY & T. ROGERS (2006): Web Services Addressing 1.0 - Core. Online unter: http://www.w3.org/TR/ws-addr-core/

 Copyright © 2006 Open Geospatial Consortium, Inc. All Rights Reserved.