

# Open Geospatial Consortium Inc.

Date: 2005-05-20

Reference number of this OGC™ IP initiative document: **OGC 05-007r2**

Version: 0.3.0

Category: OGC™ Draft Implementation Specification

Editor: Peter Schut

## Web Processing Service (WPS) Specification

### Copyright notice

Copyright ©2005 Open Geospatial Consortium, Inc

### Warning

This document is not an OGC Standard. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard.

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® Publicly Available Specification**  
Document subtype: **Discussion Paper**  
Document stage: **Final**  
Document language: **English**

Copyright © 2005 Open Geospatial Consortium, Inc.

**NOTICE**

Permission to use, copy, and distribute this document in any medium for any purpose and without fee or royalty is hereby granted, provided that you include the above list of copyright holders and the entire text of this NOTICE.

We request that authorship attribution be provided in any software, documents, or other items or products that you create pursuant to the implementation of the contents of this document, or any portion thereof.

No right to create modifications or derivatives of OGC documents is granted pursuant to this license. However, if additional requirements (as documented in the Copyright FAQ at [http://www.opengeospatial.org/about/?page=ipr&view=ipr\\_faq](http://www.opengeospatial.org/about/?page=ipr&view=ipr_faq)) are satisfied, the right to create modifications or derivatives is sometimes granted by the OGC to individuals complying with those requirements.

THIS DOCUMENT IS PROVIDED "AS IS," AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THE DOCUMENT ARE SUITABLE FOR ANY PURPOSE; NOR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE DOCUMENT OR THE PERFORMANCE OR IMPLEMENTATION OF THE CONTENTS THEREOF.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to this document or its contents without specific, written prior permission. Title to copyright in this document will at all times remain with copyright holders.

RESTRICTED RIGHTS LEGEND. Use, duplication, or disclosure by government is subject to restrictions as set forth in subdivision (c)(1)(ii) of the Right in Technical Data and Computer Software Clause at DFARS 252.227.7013

OpenGIS®, OGC™, OpenGeospatial™ and OpenLS ® are trademarks or registered trademarks of Open Geospatial Consortium, Inc. in the United States and in other countries.

<b>Contents</b>	<b>Page</b>
iv. Revision history.....	vi
v. Changes to the OGC® Abstract Specification .....	vii
Foreword.....	viii
Introduction.....	ix
<b>1 Scope.....</b>	<b>1</b>
<b>2 Conformance .....</b>	<b>1</b>
<b>3 Normative references.....</b>	<b>1</b>
<b>4 Terms and definitions.....</b>	<b>2</b>
4.1 operation .....	2
4.2 interface.....	2
4.3 service .....	2
4.4 server .....	2
4.5 client.....	2
4.6 request .....	2
4.7 response .....	2
4.8 map .....	2
4.9 capabilities.....	2
4.10 process .....	2
4.11 input.....	2
4.12 output.....	2
<b>5 Conventions .....</b>	<b>2</b>
5.1 Abbreviated terms.....	2
5.2 UML notation .....	3
5.3 Document terms and definitions .....	5
<b>6 Web Processing Service overview.....</b>	<b>5</b>
<b>7 Web Processing Service Operations.....</b>	<b>7</b>
7.1 GetCapabilities (required).....	7
7.1.1 Introduction.....	7
7.1.2 GetCapabilities request parameters .....	7
7.1.3 GetCapabilities response.....	8
7.1.3.1 Normal response.....	8
7.1.3.2 OperationsMetadata section standard contents.....	8
7.1.3.3 Capabilities document XML encoding.....	9
7.1.3.4 Exceptions.....	9
7.2 DescribeProcess operation (required) .....	9
7.2.1 Introduction.....	9
7.2.2 DescribeProcess request parameters .....	10
7.2.2.1 DescribeProcess request KVP encoding (required).....	10
7.2.2.2 DescribeProcess request XML encoding (optional).....	11
7.2.3 DescribeProcess response.....	11

- 7.2.3.1 DescribeProcess normal response .....11
- 7.2.3.2 DescribeProcess exceptions .....12
- 7.3 Execute operation (required) .....13
  - 7.3.1 Introduction.....13
  - 7.3.2 Execute request .....13
    - 7.3.2.1 Execute request parameters.....13
    - 7.3.2.2 Execute request KVP encoding (required).....14
    - 7.3.2.3 Execute request XML encoding (optional).....14
  - 7.3.3 Execute response .....15
    - 7.3.3.1 Execute normal response.....15
    - 7.3.3.2 Execute exceptions .....16
- Annex A (normative) Web Processing Service XML Schema .....18
  - A.2 List of schema files .....18
- Annex B (informative) Example XML documents.....20
  - B.1 Introduction .....20
  - B.2 List of the example XML documents.....20
- Bibliography .....21
- Questions and Options for Future work.....22

<b>Figures</b>	<b>Page</b>
<b>Figure 1 — UML notations.....</b>	<b>4</b>

<b>Tables</b>	<b>Page</b>
<b>Table 1 — GetCapabilities operation request URL parameters</b>	<b>7</b>
<b>Table 2 — Section name values and contents</b>	<b>8</b>
<b>Table 3 — Required values of OperationsMetadata section attributes</b>	<b>8</b>
<b>Table 4 — Parameters in DescribeProcess operation request</b>	<b>10</b>
<b>Table 5 — DescribeProcess operation request URL parameters</b>	<b>10</b>
<b>Table 8 — Exception codes for DescribeProcess operation</b>	<b>13</b>
<b>Table 10 — WPS Execute operation request URL parameters</b>	<b>14</b>
<b>Table 12 — Valid &lt;status&gt; codes for the status document</b>	<b>15</b>
<b>Table 13 — Exception codes for Execute operation</b>	<b>17</b>

## i. Preface

This document is the result of work undertaken to support the Canadian Geospatial Data Infrastructure (CGDI), and in particular the National Land and Water Information Service (NLWIS), and the National Forest Information Service (NFIS). The standard was first implemented as a prototype in 2004 by Agriculture and Agri-Food Canada (AAFC).

## ii. Submitting organizations

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

The following organizations submitted this Implementation Specification to the Open GIS Consortium Inc.:

GeoConnections / Natural Resources Canada

## iii. Document contributor contact points

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

Contact	Company	Address
Peter Schut	Agriculture and Agri-Food Canada	960 Carling Ave. Ottawa, ON K0A 1A0 Canada
Stephane Fellah	Image Matters LLC	
Xiaoyaun Geng	Agriculture and Agri-Food Canada	960 Carling Ave. Ottawa, ON K1A 0C6 Canada
Stephen Keens	PCI	
Weisheng Li	PCI	
Martin Kyle		
Christian Kiehle & Christian Heier		
Mike Adair		
Nicole Ostlaender & Harald Borsutzky		
Maru Newby	Agriculture and Agri-Food Canada	960 Carling Ave. Ottawa, ON K1A 0C6 Canada

## iv. Revision history

Date	Release	Editor	Primary clauses modified	Description
05 May 2004	0.1.0	P. Schut	All	Initial document, formatted for OGC template
22 May 2004	0.1.0	P. Schut	All	Cleaned up some problems, added informative examples in Annex B

21 Oct. 2004	0.2.0	Stephane Fellah	Content	Rewrite the schema and the Table of Contents
22 Nov. 2004	0.2.0	Xiaoyuan Geng	All	Created document using the latest OGC template, the initial draft, and schema
24 Dec. 2004	0.2.1	Peter Schut	All	Minor corrections and revisions throughout, additions of human readable explanations of schemas
11 April, 2005	0.2.3	Peter Schut	All	Upgrade based on results to date of WPSie.
05 May, 2005	0.3.0	Peter Schut	All	Upgrade based on results to date of WPSie and alignment with OWS Common
20 May, 2005	0.3.0	Maru Newby	All	Synchronized specification with WPS 0.3.0 changes resulting from alignment to OWS 1.1.0

#### v. **Changes to the OGC™ Abstract Specification**

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

## **Foreword**

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights. However, to date, no such rights have been claimed or identified.

The Web Processing Service (WPS) was originally named Geoprocessing Service (OGC document number 04-043).

## Introduction

This document is the specification for a Web Processing Service (WPS).

A Web Service Processing Service provides access to calculations or models which operate on spatially referenced data. The data required by the service can be available locally, or delivered across a network using data exchange standards such as Geography Markup Language (GML) or Geolinked Data Access Service (GDAS). The calculation can be as simple as subtracting one set of spatially referenced numbers from another (e.g. determining the difference in influenza cases between two different seasons), or as complicated as a global climate change model.

This specification is intended to provide a mechanism to identify the spatially-referenced data required by the calculation, initiate the calculation, and manage the output from the calculation so that it can be accessed by the client. The Web Processing Service is targeted at both vector and raster data based processing.



# OpenGIS<sup>®</sup> Web Processing Service (WPS) Specification

## 1 Scope

This specification applies to the creation and use of a Web Processing Service which exposes pre-programmed calculations for geospatial data to the Internet.

This specification does not address the archival, cataloguing, discovery or retrieval of information that has been created by a web processing service.

## 2 Conformance

Conformance with this specification shall be checked using all the relevant tests specified in Annex A (normative). The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in ISO 19105: Geographic information — Conformance and Testing.

## 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 05-008c1, OWS Common Implementation Specification, June 2004

[1] CGDI architecture pages at <http://www.geoconnections.org/architecture/>

[2] XML 1.0 (October 2000), *eXtensible Markup Language (XML) 1.0 (2nd edition)*, World Wide Web Consortium Recommendation, Bray, T., Paoli, J., Sperberg-McQueen, C.M., and Maler, E., eds., <<http://www.w3.org/TR/2000/REC-xml>>

[3] IETF RFC 2045 (November 1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, Freed, N. and Borenstein N., eds., <http://www.ietf.org/rfc/rfc2045.txt>

[4] IETF RFC 2119 (March 1997), *Key words for use in RFCs to Indicate Requirement Levels*, Bradner, S., ed., <<http://www.ietf.org/rfc/rfc2119.txt>>.

## 4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

### 4.1 operation

specification of a transformation or query that an object may be called to execute [OGC AS 12]

### 4.2 interface

named set of **operations** that characterize the behavior of an entity [OGC AS 12]

### 4.3 service

distinct part of the functionality that is provided by an entity through **interfaces** [OGC AS 12]

### 4.4 server

actual implementation of a **service**

### 4.5 client

software component that can invoke an **operation** from a **server**

### 4.6 request

invocation of an **operation** by a **client**

### 4.7 response

result of an **operation** returned from a **server** to a **client**

### 4.8 map

pictorial representation of geographic data.

### 4.9 capabilities

service-level metadata describing the **operations** and content available at a **service instance**.

### 4.10 process

model or calculation that is made available at a **service instance**.

### 4.11 input

data provided to a **process**.

### 4.12 output

result returned by a **process**.

## 5 Conventions

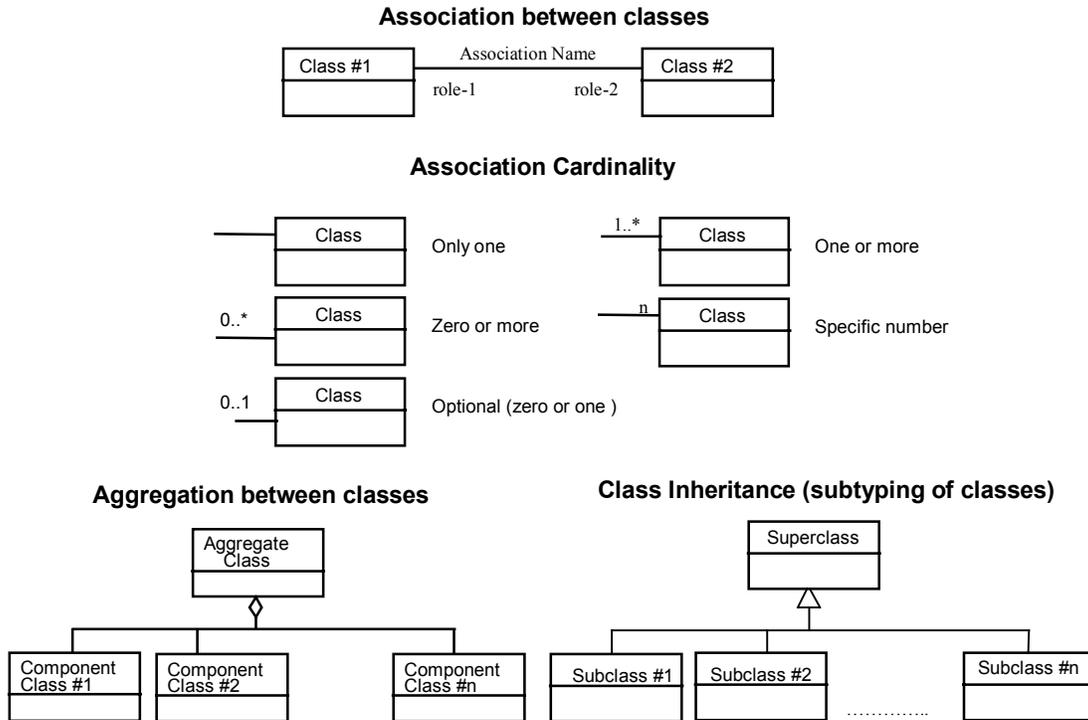
### 5.1 Abbreviated terms

API	Application Program Interface
CGDI	Canadian Geospatial Data Infrastructure

CGI	Common Gateway Interface
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off The Shelf
DCE	Distributed Computing Environment
DCOM	Distributed Component Object Model
EPSG	European Petroleum Survey Group
GIF	Graphics Interchange Format
GIS	Geographic Information System
HTTP	Hypertext Transfer Protocol
IDL	Interface Definition Language
IETF	Internet Engineering Task Force
JPEG	Joint Photographic Experts Group
MIME	Multipurpose Internet Mail Extensions
OGC	Open GIS Consortium
OWS	OGC Web Service
PNG	Portable Network Graphics
RFC	Request for Comments
SLD	Styled Layer Descriptor
URL	Uniform Resource Locator
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
XML	Extensible Markup Language

## 5.2 UML notation

Some of the diagrams in this document are presented using the Unified Modeling Language (UML) static structure diagram. The UML notations used in this document are described in the diagram below.



**Figure 1 — UML notations**

In these UML class diagrams, the class boxes with a light background are the primary classes being shown in this diagram, often the classes from one UML package. The class boxes with a gray background are other classes used by these primary classes, usually classes from other packages.

In this diagram, the following stereotypes of UML classes are used:

- <<DataType>> A descriptor of a set of values that lack identity (independent existence and the possibility of side effects). A DataType is a class with no operations, whose primary purpose is to hold the information.
- a) <<Enumeration>> A data type whose instances form a list of alternative literal values. Enumeration means a short list of well-understood potential values within a class.
- b) <<CodeList>> A flexible enumeration for expressing a long list of potential alternative values. If the list alternatives are completely known, an enumeration shall be used; if the only likely alternatives are known, a code list shall be used.
- c) <<Interface>> A definition of a set of operations that is supported by objects having this interface. An Interface class cannot contain any attributes.
- d) <<Type>> A stereotyped class used for specification of a domain of instances (objects), together with the operations applicable to the objects. A Type class may have attributes and associations.
- e) <<Union>> A list of alternate attributes where only one of those attributes can be present at any time.

NOTE All the stereotypes listed above are adapted from Subclauses 6.8.2 and D.8.3 of ISO 19103.

In this document, the following standard data types are used:

CharacterString – A sequence of characters

Boolean – A value specifying TRUE or FALSE

URI – An identifier of a resource that provides more information

URL – An identifier of an on-line resource that can be electronically accessed

Integer – An integer number

Double – A double precision floating point number

### 5.3 Document terms and definitions

The following specification terms and definitions are used in this document:

shall – verb form used to indicate a requirement to be strictly followed to conform to this specification, from which no deviation is permitted

f) should – verb form used to indicate desirable ability or use, without mentioning or excluding other possibilities

g) may – verb form used to indicate an action permissible within the limits of this specification

h) can – verb form used for statements of possibility

i) informative – a part of a document that is provided for explanation, but is not required

j) normative – a part of a standards document that is required

k) annex – an auxiliary part of a document, called an “appendix” in United States English

l) clause – a major part of a document, called a “section” or “paragraph” in United States English

m) subclause – a secondary part of a clause or annex, called a “subsection” in United States English

## 6 Web Processing Service overview

The Web Processing Service (WPS) interface specifies WPS operations that can be requested by a client and performed by a WPS server. Those operations are:

1) GetCapabilities – This operation allows a client to request and receive back service metadata (or Capabilities) documents that describe the abilities of the specific server implementation, including the names of the processes which can be executed. This operation also supports negotiation of the specification version being used for client-server interactions.

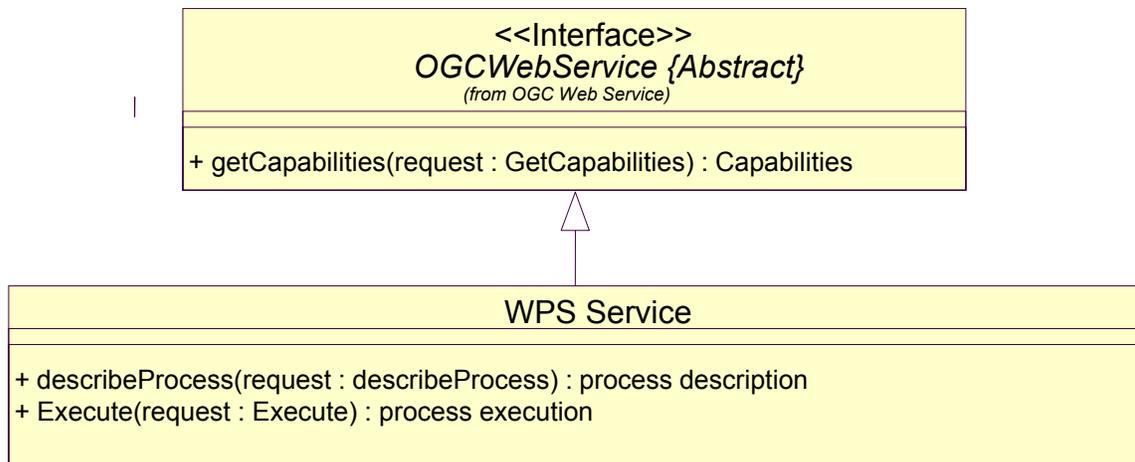
2) DescribeProcess – This operation allows a client to acquire more specific information

about an Execute operation provided by the WPS, including the input parameters and formats, and the outputs.

3) Execute – This operation allows a client to run a specified process of the WPS with qualified input parameters and values.

These operations have many similarities to other OGC Web Services, including the WMS, WFS, and WCS. Many of these interface aspects that are common with other OWSs are thus specified in the OGC Web Services Common Implementation Specification [OGC 05-008c1]. These common aspects are normatively referenced herein, instead of being repeated in this specification.

Figure 1 is a simple UML diagram summarizing the WPS interface. This class diagram shows that the WPS interface class inherits the getCapabilities operation from the OGCWebService interface class, and adds the describeProcess and Execute operations. (This capitalization of names uses the OGC/ISO profile of UML.) A more complete UML model of the WPS is provided in Annex D (informative).



Each server instance instantiates only one object of this class, and this object always exists while server is available.

**Figure 2 — WPS interface UML diagram**

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

## 7 Web Processing Service Operations

The WPS operations are defined by a set of XML schemas, which are completely annotated, and document the meaning of each element, and their allowable attributes and type. The documentation in these schemas shall be considered normative as specified in Subclause 11.6.3 of [OGC 05-008c1]. This section describes the purpose of each operation and provides insight into the appropriate use that can not be accommodated readily in the schemas.

### 7.1 GetCapabilities (required)

#### 7.1.1 Introduction

This clause partially specifies the GetCapabilities operation provided by each OWS. The mandatory GetCapabilities operation allows any client to retrieve metadata about the services available from any server that implements an OWS interface Implementation Specification. The normal response to the GetCapabilities operation is a service metadata document that is returned to the requesting client. This service metadata document primarily contains metadata about the specific server abilities (such as about the specific data and formats available from that server). This service metadata also makes an OWS server partially self-describing, supporting late binding of clients.

**NOTE** A specific OWS Implementation Specification or implementation can provide additional operation(s) returning service metadata for a server. Such operations can return service metadata using different data structures and/or formats, such as WSDL or eBRIM. When such operation(s) have been sufficiently specified and shown more useful, the OGC may decide to require those operation(s) instead of the current GetCapabilities operation.

#### 7.1.2 GetCapabilities request parameters

The GetCapabilities operation request shall be as specified in Subclauses 7.2.2 through 7.2.4 of [OGC 05-008c1]. The “service”, “request”, and “AcceptVersions” parameters shall be implemented by all WPS servers. The “Sections” and “updateSequence” parameters are optional implementation by WPS servers. All WPS servers shall implement HTTP GET transfer of the GetCapabilities operation request, using KVP encoding. Servers can also implement HTTP POST transfer of the GetCapabilities operation request, using XML encoding.

**Table 1 — GetCapabilities operation request URL parameters**

Name and example <sup>a</sup>	Optionality and use	Definition and format
service=WPS	Mandatory	Service type identifier
request= GetCapabilities	Mandatory	Operation name
Version=0.3.0	Optional	Specification and schema version for this operation

An example GetCapabilities request for HTTP GET / KVP encoding is shown below:

```

http://foo.bar/foo?
  service="WPS"&
  version="0.3.0"&
  request="GetCapabilities"

```

### 7.1.3 GetCapabilities response

#### 7.1.3.1 Normal response

The service metadata document shall contain the sections specified in Table 2. Depending on the values in the Sections parameter of the GetCapabilities operation request, any combination of these sections can be requested and returned.

**Table 2 — Section name values and contents**

Section name	Contents
ServiceIdentification	Metadata about this specific server. The schema of this section shall be the same as for all OWSs, as specified in Subclause 7.4.3 and owsServiceIdentification.xsd of [OGC 05-008c1].
ServiceProvider	Metadata about the organization operating this server. The schema of this section shall be the same for all OWSs, as specified in Subclause 7.4.4 and owsServiceProvider.xsd of [OGC 05-008c1].
OperationsMetadata	Metadata about the operations specified by this service and implemented by this server, including the URLs for operation requests. The basic contents and organization of this section shall be the same as for all OWSs, as specified in Subclause 7.4.5 and owsOperationsMetadata.xsd of [OGC 05-008c1].

In addition to these sections, each service metadata document shall include the mandatory “version” and optional updateSequence parameters specified in Table 6 in Subclause 7.4.1 of [OGC 05-008c1].

#### 7.1.3.2 OperationsMetadata section standard contents

For the WPS, the OperationsMetadata section shall be the same as for all OGC Web Services, as specified in Subclause 7.4.5 and owsOperationsMetadata.xsd of [OGC 05-008c1]. The mandatory values of various (XML) attributes shall be as specified in Table 3. The “Attribute name” column uses dot-separator notation to identify parts of a parent item. The “Attribute value” column references an operation parameter, in this case an operation name, and the meaning of including that value is listed in the right column.

**Table 3 — Required values of OperationsMetadata section attributes**

Attribute name	Attribute value	Meaning of attribute value
Operation.name	GetCapabilities	The GetCapabilities operation is implemented by this server.
	DescribeProcess	The DescribeProcess operation is implemented by this server.
	Execute	The Execute operation is implemented by this server.

### 7.1.3.3 Capabilities document XML encoding

The WPS GetCapabilities.xsd schema extends ows:CapabilitiesBaseType in owsCommon.xsd of [OGC 05-008c1]. The schema uses the owsServiceIdentification.xsd, owsServiceProvider.xsd, and owsOperationsMetadata.xsd schemas specified in [OGC 05-008c1].

Specific to WPS is the ProcessOfferings section which contains the names of one or more processes that can be executed by the service, and a general description of each process. In order to obtain more detailed information about a process, the DescribeProcess operation is used. An example of a ProcessOfferings section is shown in the following XML schema fragment.

```
<wps:ProcessOfferings>
  <wps:Process>
    <ows:Identifier>intersection</ows:Identifier>
    <ows:Title>Intersection of one GML polygon with another</ows:Title>
    <ows:Abstract>Intersect the polygon coordinates found in one GML stream with a second
    GML stream, where both streams are in the same coordinate reference space. Output is a
    GML polygon feature that represents the area of intersection. Supports GML formats up to
    version 2.0.0.</ows:Abstract>
    <wps:Version>1.0</wps:Version>
    <wps:Application>
      <ows:Keywords>
        <ows:Keyword>Topological</ows:Keyword>
      </ows:Keywords>
    </wps:Application>
    <wps:Category>
      <ows:Keywords>
        <ows:Keyword>Intersection</ows:Keyword>
        <ows:Keyword>Polygon</ows:Keyword>
      </ows:Keywords>
    </wps:Category>
  </wps:Process>
</wps:ProcessOfferings>
```

### 7.1.3.4 Exceptions

When a WPS server encounters an error while performing a GetCapabilities operation, it shall return an exception report message as specified in Clause 8 of [OGC 05-008c1]. The allowed exception codes shall include those listed in Table 5 of Subclause 7.4.1 of [OGC 05-008c1], if the updateSequence parameter is implemented by the server.

## 7.2 DescribeProcess operation (required)

### 7.2.1 Introduction

The DescribeProcess operation allows WPS clients to get a full description of one or more processes to be invoked. It defines the inputs and outputs of the process with their respective domain of validity. This description can be used to build an automatic user interface to capture the parameters values used to execute a process instance.

### 7.2.2 DescribeProcess request parameters

A request to perform the DescribeProcess operation shall include the parameters listed and defined in Table 4. This table also specifies the UML model data type, source of values, and multiplicity of each listed parameter, plus the meaning to servers when each optional parameter is not included in the operation request. Although some values listed in the “Name” column appear to contain spaces, they shall not contain spaces.

NOTE 1 To reduce the need for readers to refer to other documents, the first three parameters listed below are copied from Table 21 in Subclause 9.2.1 of [OGC 05-008c1].

**Table 4 — Parameters in DescribeProcess operation request**

Name <sup>a</sup>	Definition	Data type and value	Multiplicity and use
service	Service type identifier	WPS	One (mandatory)
request	Operation name	DescribeProcess	One (mandatory)
version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
Identifier	Process identifiers list	Character String type, not empty Value is process name (e.g., “addition”). Process names as defined in ProcessOfferings section in the capability document.	One or more (mandatory).

a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008c1].

NOTE 2 The data type of many parameters is specified as “Character String type, not empty”. In the XML Schemas specified herein, these parameters are encoded with the xsd:string type, which does NOT require that these strings not be empty.

#### 7.2.2.1 DescribeProcess request KVP encoding (required)

Servers shall implement HTTP GET transfer of the WPS operation request, using KVP encoding. The KVP encoding of the WPS operation request shall use the parameters specified in Table 5. The parameters listed in Table 5 shall be as specified in Table 4 above.

**Table 5 — DescribeProcess operation request URL parameters**

Name and example <sup>a</sup>	Optionality and use	Definition and format
service=WPS	Mandatory	Service type identifier
request= DescribeProcess	Mandatory	Operation name
Version=0.3.0	Mandatory	Specification and schema version for this operation
Identifier=	One (mandatory) or more	Process identifier as shown in the capabilities document (separated by commas)

A All parameter names are here listed using mostly lower case letters. However, any parameter name capitalization shall be allowed in KVP encoding, see Subclause 11.5.2 of [OGC 05-008c1].

### 7.2.2.2 DescribeProcess request XML encoding (optional)

It is optional for WPS servers to implement HTTP POST transfer of the WPS operation request, using XML encoding . A DescribeProcess operation request for WPS can look like this encoded in XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<DescribeProcess service="WPS" version="0.3.0" xmlns="http://www.opengis.net/wps"
xmlns:ows="http://www.opengeospatial.net/ows">
  <ows :Identifier>subtraction</ows :Identifier>
  <ows:Identifier>addition</ows:Identifier>
</DescribeProcess>
```

### 7.2.3 DescribeProcess response

#### 7.2.3.1 DescribeProcess normal response

The normal response to a DescribeProcess request is at least one or more Process Descriptions. The values of various (XML) attributes shall be as specified in the wpsProcess.xsd XML schema. This schema contains annotations that completely describe each element.

The response to a DescribeProcess request includes the information returned in the GetCapabilities response, plus the input and output parameters. There can be any number of input and output parameters. Each parameter is described within a <Parameter> element, and the allowable formats and units of measure are specified.

Essentially, the DescribeProcess response tells the client one or more of the following types of statements:

1. "I need a <Reference> URL that contains a file corresponding to the following gml:remoteSchema" (e.g geoTIFF, GML 0.3.1, GDAS 0.9.1, etc.)
2. "I need a <LiteralValue> parameter, with the following <AllowedValues>, <DefaultValue> and <SupportedUoms>."
3. "I need a <ComplexValue> which must be in one of the following <format>s".
4. "I need a <BoundingBox>"

Note that complex data inputs can be handled in two ways: either by using a <datatype> of type <reference> to point to a URL that contains the required input, or by embedding the content directly into the request using <ComplexValue>.

An example response to a DescribeProcess operation request is:

```
<?xml version="1.0" encoding="UTF-8"?>
<ProcessDescription version="0.3.0" xmlns="http://www.opengis.net/wps"
xmlns:ows="http://www.opengeospatial.net/ows">
  <Process>
```

```

<ows:Identifier>GmlIntersectToGml</ows:Identifier>
<ows:Name>2 GML stream intersected and result will be returned in GML stream</ows:Name>
<ows:Abstract>Intersect two GML format 2.1.2 and return a GML result</ows:Abstract>
<Version>2</Version>
<Application>
  <ows:Keywords>
    <ows:Keyword>Spatial</ows:Keyword>
    <ows:Keyword>Operation</ows:Keyword>
    <ows:Keyword>geometry</ows:Keyword>
  </ows:Keywords>
</Application>
<Category>
  <ows:Keywords>
    <ows:Keyword>Intersect</ows:Keyword>
    <ows:Keyword>GML</ows:Keyword>
  </ows:Keywords>
</Category>
<Input>
  <Parameter>
    <ows:Identifier>HRefIntersectTo</ows:Identifier>
    <ows:Title>HRefIntersectTo</ows:Title>
    <ows:Abstract>URI to a GML resource file</ows:Abstract>
    <MinimumOccurs>1</MinimumOccurs>
    <Reference remoteSchema="http://schemas.opengis.net/gml/2.1.2"/>
  </Parameter>
</Input>
<Input>
  <Parameter>
    <ows:Identifier>HRef</ows:Identifier>
    <ows:Title >HRef</ows:Title >
    <ows:Abstract>URI to a GML resource file</ows:Abstract>
    <MinimumOccurs>1</MinimumOccurs>
    <Reference remoteSchema="http://schemas.opengis.net/gml/2.1.2"/>
  </Parameter>
</Input>
<Output>
  <Parameter>
    <ows:Identifier>Intersection</ows:Identifier>
    <ows:Title>Intersection</ows:Title >
    <ows:Abstract>A GML format stream will be returned to the client or an Exception Error
if nothing found</ows:Abstract>
    <MinimumOccurs>1</MinimumOccurs>
    <ComplexValue remoteSchema="http://schemas.opengis.net/gml/2.1.2"/>
  </Parameter>
</Output>
</Process>
</ProcessDescription>

```

### 7.2.3.2 DescribeProcess exceptions

When a WPS server encounters an error while performing a DescribeProcess operation, it shall return an exception report message as specified in Subclause 7.4 of [OGC 05-008c1]. The allowed standard exception codes shall include those listed in Table 8. For each listed exceptionCode, the contents of the “locator” parameter value shall be as specified in the right column of Table 8.

NOTE To reduce the need for readers to refer to other documents, the first four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008c1].

**Table 8 — Exception codes for DescribeProcess operation**

<b>exceptionCode value</b>	<b>Meaning of code</b>	<b>“locator” value</b>
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	Name of parameter with invalid value
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit “locator” parameter

### 7.3 Execute operation (required)

#### 7.3.1 Introduction

The Execute operation allows WPS clients to execute a Process instance with a list of given parameters. This operation requires that the inputs be identified, as defined in the Process Description. If there is a single output to the process, it can be returned as a direct response to the request. Alternatively, the server can be directed to store the result as a web accessible resource. If there are multiple outputs, the results must be stored as a set of web-accessible resources.

If the results are stored, the direct response includes a <status> element, which includes information about the status of the process, as well as a status URL which is used to obtain updated information about the status of the process. This URL is useful for processes that take a substantial amount of time to execute. The location of the outputs are also identified in the <status> element, so the client can retrieve them when the process has completed.

[Put in Workflow diagrams to clarify what happens when output is stored!!!](#)

#### 7.3.2 Execute request

##### 7.3.2.1 Execute request parameters

A request to perform the WPS Execute operation shall include the parameters listed and defined in the wpsExecute.xsd schema. This schema also specifies the UML model data type, source of values, and multiplicity of each parameter, plus the meaning to servers when each optional parameter is not included in the operation request.

The normal way to provide inputs to a WPS is through the identification of one or more remote URIs, unless the parameter is a simple scalar value.

The schema provides support for multiple inputs. These inputs refer to separate types of input that may be required for a single Execute request. This is not intended to be used to facilitate batch processing (e.g. multiple images to be processed through a single algorithm)

[clarify how ComplexValue is used, and then give some examples \(Roberto suggested GML and Base64\)](#)

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

### 7.3.2.2 Execute request KVP encoding (required)

Servers must implement HTTP GET/POST transfer of the WPS operation request, using KVP encoding. The KVP encoding of the WPS operation request shall use the parameters specified in Table 10.

**Table 10 — WPS Execute operation request URL parameters**

Name and example <sup>a</sup>	Optionality and use	Definition and format
service=WPS	Mandatory	Service type identifier
request= Execute	Mandatory	Operation name
Version= 0.3.0	Mandatory	Specification and schema version for this operation
ProcessName	Mandatory	Character String type. Valid contents are based on the name of the process as identified in the DescribeProcess response
Store	Optional	Character String type: “true” or “false”, where default is false
ParameterName <sup>b</sup>	Mandatory, more than one allowed.	Character String type, not empty Value is specified by each Implementation of the Specification
<p>a All parameter names are here listed using mostly lower case letters. However, any parameter name capitalization shall be allowed in KVP encoding, see Subclause 11.5.2 of [OGC 05-008c1].</p> <p>b. Dependent upon the parameter name, “ParameterName” should be substituted by each of the names of the required input parameters identified in the DescribeProcess response.</p>		

An example Execute request using KVP Get is shown below:

```

http://foo.bar/foo?
  service="WPS"&
  request="Execute"&
  version="0.3.0"&
  store="true"&
  ProcessName="Intersection"&
  polygon1="http://foo.bar/foo1"&
  polygon2="http://foo.bar/foo2"

```

### 7.3.2.3 Execute request XML encoding (optional)

WPS servers may implement HTTP POST transfer of the WPS Execute request, using XML encoding. The wpsExecute.xsd schema specifies the contents and structure of a

WPS Execute request encoded in XML. An example of a WPS operation request for “execute” can look like this encoded in XML:

[insert example XML fragment here](#)

### 7.3.3 Execute response

#### 7.3.3.1 Execute normal response

The schema allows a WPS to provide multiple outputs as a result of an Execute request. In the case of an indirect (stored) request, this is supported in an ExecuteResponse element by identifying a separate URL for each of the multiple output parameters. Multiple outputs are not supported when a "direct" response is requested.

The default (i.e Output.Store=”False”) response to a WPS request returns the output from the process directly to the client. For example, if a WPS process is supposed to create GML as its output, then that GML will be returned to the client as a direct response to the Execute request.

If storage of the output has been requested (i.e. Output.Store=“True”), the Execute response complies with the ExecuteResponse element of the wps.Execute.xsd schema. WPS is not specifically designed to store outputs for the long term. Clients are expected to download the outputs to some other web-accessible location if long term storage is required.

An example response to an Execute operation request where “store” is true is:

[insert example XML fragment here](#)

Note that this response includes the StatusReport element. This element contains a URL that will return both status information about the Execute request, and the URL(s) at which the output(s) may be retrieved. If the process has not completed by the time the response is sent, the location(s) of the output(s) will not necessarily be identified.

Essentially the same content is available from the status URL. This URL returns XML that complies with the wps.Status.xsd schema. The major content of the status document is the <status> element. The possible values of status are shown in Table 12 below.

**Table 12 — Valid <status> codes for the status document**

Value of <Status>	Meaning and content
PROCESS_ACCEPTED	The WPS has accepted the input and has queued it for processing
PROCESS_STARTED	The WPS has started processing the request. An estimated execution time sub element is filled in and updated by the server at the discretion of the developer.

PROCESS_SUCCESS	The WPS has completed the process One or more URL element(s) are included which reference the actual result(s).
PROCESS_FAILED	The WPS process failed. An error code sub element identifies the problem encountered

Once a process has completed successfully, <status> must take on the value of “PROCESS\_SUCCESS”, and the reference URL(s) to the output(s) must be populated.

An example status document is shown below:

**Insert example XML fragment here**

The URL(s) of the output(s) shall be populated with the results of the process as soon as they are available. If the URL(s) are accessed before the process has had time to populate the online resource(s), or the storage time for the resources has been exceeded, the server shall return an Error 403 (Not Found).

The content of the execute response (and status document) contains the input that was provided by the client, in the <input> section. This includes any URI provided in the execute request. If the input was embedded in the request, then the server may generate and populate one or more URLs for the payload.

### 7.3.3.2 Execute exceptions

When a WPS server encounters an error while performing a WPS operation, it shall return an exception report message as specified in Subclause 7.4 of [OGC 05-008c1]. The allowed standard exception codes shall include those listed in Table 12. For each listed exceptionCode, the contents of the “locator” parameter value shall be as specified in the right column of Table 13.

NOTE To reduce the need for readers to refer to other documents, the first four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008c1].

**Table 13 — Exception codes for Execute operation**

<b>exceptionCode value</b>	<b>Meaning of code</b>	<b>“locator” value</b>
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	Name of parameter with invalid value
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit “locator” parameter

## Annex A (normative)

### Web Processing Service XML Schema

#### A.1 WPS xml schema and service exception xml schema

In addition to this document, this specification includes several normative XML Schema files. These are posted online at the URL <http://testschemas.openeospatial.net/WPSSchemaRepository/wps/> where a lower level directory is used for this version of WPS. These XML Schema files are also bundled in a zip file with the present document. In the event of a discrepancy between the bundled and online versions of the XML Schema files, the online files shall be considered authoritative.

- In the event of a discrepancy between the text in this document and the schemas, the schemas shall be considered authoritative.
- In the event of a discrepancy between the bundled and online versions of the XML example files, the online files shall be considered authoritative

#### A.2 List of schema files

These XML Schema files used for this specification are:

wpsCommon.xsd  
wpsDescribeProcess.xsd  
wpsExecute.xsd  
wpsGetCapabilities.xsd  
wpsProcess.xsd  
wpsStatus.xsd

These XML Schemas use and build on the OWS common XML Schemas specified [OGC 05-008c1], named:

owsServiceIdentification.xsd  
owsDataIdentification.xsd  
ows19115subset.xsd  
owsServiceProvider.xsd  
owsOperationsMetadata.xsd  
owsExceptionReport.xsd  
owsBoundingBox.xsd  
owsGetCapabilities.xsd

All these XML Schemas contain documentation of the meaning of each element and attribute, and this documentation shall be considered normative as specified in Subclause 11.6.3 of [OGC 05-008c1].

The WPS XML Schemas also depend on the following schemas:

- parameterValues.xsd
- referenceTypeUsingGML.xsd

parameterValues.xsd has been refactored to owsDomainType.xsd in OWS specified [OGC 05-008c1], but functionality that the WPS depends on has been removed from owsDomainType.xsd. referenceTypeUsingGML.xsd is not currently part of OWS, but will likely be included in future releases. These unofficial/unsupported schema files will be distributed with the core WPS schema files until their functionality can be refactored into WPS or incorporated into a release of the OWS schemas.

## **Annex B** (informative)

### **Example XML documents**

#### **B.1 Introduction**

In addition to this document, this specification includes several XML example files. These are posted online at the URL <http://portal.opengeospatial.org/wiki/twiki/bin/view/WPSie/WebHome> where a lower level directory is used for this Version of WPS. These XML example files are also bundled in a zip file with the present document.

#### **B.2 List of the example XML documents**

wpsDescribeProcess.xml  
wpsDescribeProcessResponse.xml

wpsExceptionReport.xml

wpsExecute.xml  
wpsExecuteResponse.xml

wpsGetCapabilitiesResponse.xml

## Bibliography

- [1] ISO 31 (all parts), Quantities and units.
- [2] IEC 60027 (all parts), Letter symbols to be used in electrical technology.
- [3] ISO 1000, SI units and recommendations for the use of their multiples and of certain other units.
- [4] [4] Guidelines for Successful OGC Interface Specifications, OGC document 00-014r1

## Questions and Options for Future work

1. Should we explicitly support different input and output versions for `input.parameter.datatype` and `output.parameter.datatype` (e.g. GML 2.2 and GML 3.0)? Although we don't have to - because a service could support different data format versions by providing separate processes... it would be more efficient.
2. Should the execute POST option really be mandatory?
3. Having a definitive list of keywords to be used in "Application" and "Category" would be useful, but maintaining them in the spec is problematic – it would be better if this list were to be maintained separately in RDF on the OGC website.
4. Similarly, in terms of promoting interoperability, it is critical that it be possible to elucidate the meanings of inputs and outputs from their names. If these names were defined on a shared site using RDF or a registry.
5. Version 0.1.0 included the capability to identify local datasets that could be used as input to a process. 0.2.1 allows a local payload to be used (Execute.xsd line 52: the `xlink:href` must start with the prefix `cid:`). The ability to publish local datasets needs to be reinstated, or an alternative described.
6. We have Name, Label, Description and Abstract to describe inputs in the Execute operation. Surely we don't need all four.
7. Standardize the capitalization of the XML tags.
8. Execute needs to be a named operation in the GetCapabilities. It is not shown in the schema or examples.
9. The output URL should contain a status report for lengthy processes. It could look something like this:
 

```

      <ProcessingStatus>
        <ProcessId>1312</ProcessId>
        <Status>processing</Status>           [pending/processing/complete]
        <StatusAsOf>2004.06.04.19:01.22</StatusAsOf>
        <EstimatedStartTime>Not Applicable</EstimatedStartTime>
        <StartTime>2004.06.04.18:03.49</StartTime>
        <ElapsedTime>0.0.0.49:02.27</ElapsedTime>   [seconds?]
        <EstimatedProcessingTime></EstimatedProcessingTime>
      </ProcessingStatus>
      
```
10. Should the label/description be balanced between the execute request and response? There may be some value in having the WPS pass them through as part of the response, and allowing the client to describe the output.
11. Fix up the `allowedValues` element within `Process.Input.Parameter`.