

**OGC® DOCUMENT: 14-065R2**

External identifier of this OGC® document: <http://www.opengis.net/doc/IS/wps/2.0.2>

**OGC®**  
Making location count.

# OGC® WPS 2.0.2 INTERFACE STANDARD CORRIGENDUM 2

---

**STANDARD**  
**Implementation**

**APPROVED**

**Version:** 2.0.2

**Submission Date:** 2014-11-25

**Approval Date:** 2015-01-27

**Publication Date:** 2015-03-05

**Editor:** Matthias Mueller, Benjamin Pross

**Notice:** This document is an OGC Member approved international standard. This document is available on a royalty free, non-discriminatory basis. 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.

### **License Agreement**

Use of this document is subject to the license agreement at <https://www.ogc.org/license>

Suggested additions, changes and comments on this document are welcome and encouraged. Such suggestions may be submitted using the online change request form on OGC web site: <http://ogc.standardstracker.org/>

### **Copyright notice**

Copyright © 2015 Open Geospatial Consortium

To obtain additional rights of use, visit <https://www.ogc.org/legal>

### **Note**

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

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

# CONTENTS

I. ABSTRACT .....	vi
II. KEYWORDS .....	vi
III. PREFACE .....	vii
IV. SECURITY CONSIDERATIONS .....	viii
V. SUBMITTING ORGANIZATIONS .....	ix
VI. SUBMITTERS .....	ix
VII. INTRODUCTION .....	ix
1. SCOPE .....	2
2. CONFORMANCE .....	4
3. NORMATIVE REFERENCES .....	6
4. TERMS AND DEFINITIONS .....	8
6. CONVENTIONS .....	11
6.1. Abbreviated terms .....	11
6.2. Use of the Term “Process” .....	11
6.3. UML Notation .....	12
6.4. Namespace Conventions .....	12
7. WPS CONCEPTUAL MODEL .....	14
7.1. Service Discovery .....	14
7.2. Service Capabilities .....	15
8. WPS NATIVE PROCESS MODEL .....	17
8.1. Common Description Type .....	17
8.2. Data Description Structure .....	19
9. WPS NATIVE PROCESS MODEL ENCODING .....	22
9.1. XML Schema Implementation .....	22
9.2. Plain Text Encoding for LiteralData and BoundingBoxData Values .....	24
10. COMMON WPS SERVICE MODEL .....	27

10.1. Overview of WPS Core Operations .....	27
10.2. Data Transmission .....	29
ANNEX A (NORMATIVE) ABSTRACT TEST SUITE .....	33
A.1. Basic WPS (Conformance Class) .....	33
ANNEX B (INFORMATIVE) XML EXAMPLES .....	35
B.1. Data Types .....	35
BIBLIOGRAPHY .....	38

## LIST OF TABLES

---

Table 1 – Conformance classes related to WPS 2.0 .....	4
Table 2 – Properties of the DescriptionType structure .....	18
Table 3 – Properties of the Metadata structure .....	19
Table 4 – Format properties .....	20
Table 5 – Parts of the inline Data structure .....	29
Table 6 – Properties of the DataEncodingAttributes structure .....	30
Table 7 – Parts of the Reference structure .....	30
Table 8 – Parts of the RequestBody structure .....	31
Table 9 – Parts of the BodyReference structure .....	31

## LIST OF FIGURES

---

Figure 1 – Artifacts of the WPS service model .....	14
Figure 2 – DescriptionType for processes, process inputs and process outputs UML class diagram .....	18
Figure 3 – DataDescription and supported formats UML class diagram .....	19
Figure 4 – Common sequence of WPS operations UML sequence diagram .....	28
Figure 5 – Input and output data transmission structures UML class diagram .....	29

## LIST OF RECOMMENDATIONS

---

REQUIREMENTS CLASS 1 .....	14
REQUIREMENTS CLASS 2 .....	15

REQUIREMENTS CLASS 3 .....	15
REQUIREMENTS CLASS 4 .....	17
REQUIREMENTS CLASS 5 .....	17
REQUIREMENTS CLASS 6 .....	19
REQUIREMENTS CLASS 7 .....	22
REQUIREMENTS CLASS 8 .....	22
REQUIREMENTS CLASS 9 .....	23
REQUIREMENTS CLASS 10 .....	23
REQUIREMENTS CLASS 11 .....	24
REQUIREMENTS CLASS 12 .....	27
REQUIREMENTS CLASS 13 .....	29
REQUIREMENT A.1 .....	33
RECOMMENDATION A.1 .....	33
REQUIREMENT A.2 .....	33

# I

## ABSTRACT

---

In many cases geospatial or location data, including data from sensors, must be processed before the information can be used effectively. The OGC Web Processing Service (WPS) Interface Standard provides a standard interface that simplifies the task of making simple or complex computational processing services accessible via web services. Such services include well-known processes found in GIS software as well as specialized processes for spatio-temporal modeling and simulation. While the OGC WPS standard was designed with spatial processing in mind, it can also be used to readily insert non-spatial processing tasks into a web services environment.

# II

## KEYWORDS

---

The following are keywords to be used by search engines and document catalogues.

TBD, geoprocessing, ogcdoc, OGC document, processes, processing, WPS



## PREFACE

---

This standard is a continuation of WPS 1.0, a standard for web-based processing of geospatial data. It incorporates a range of change requests that have been submitted since the release of WPS 1.0 and further follows the OGC standard for modular specifications [OGC 08-131r3].

In contrast to the prior version, WPS 2.0 provides a core conceptual model that may be used to specify a WPS in different architectures such as REST or SOAP.

The WPS process model has been encapsulated into separate requirements and conformance classes, so it may be used independently from WPS servers in process catalogs and metadata records. The expressive power of process descriptions has been enhanced by permitting structured (or nested) inputs and outputs. The concept of process profiles has been clarified and extended to support process descriptions at different levels of abstraction.

Conversely, the process model itself has been largely decoupled from the WPS protocol, allowing the use of other domain-specific descriptions of processes, e.g. those defined in SensorML, and to execute them on a WPS server.

This specification also provides a Basic WPS conformance class that comprises the synchronous and asynchronous execution protocol, the WPS process model, and implements HTTP/POST+XML and HTTP/GET+KVP encodings.

Future work will target the definition of process interfaces for common processes based on the process model conformance class. Such profiles will encourage the development of well-defined, reliable, interoperable and exchangeable process implementations.

If OGC baseline and related specifications should further progress towards REST-oriented interfaces, the development of a REST-oriented WPS interface standard should be considered.



## SECURITY CONSIDERATIONS

---

No security considerations have been made for this document.



# V

## SUBMITTING ORGANIZATIONS

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

- TU Dresden
- 52°North
- Intergraph Corporation
- Image Matters LLC
- CRP Henri Tudor
- Airbus Defence and Space

# VI

## SUBMITTERS

NAME	REPRESENTING	OGC MEMBER
Matthias Müller	TU Dresden	Yes
Benjamin Pross	52°North	Yes
Stan Tillman	Intergraph Corporation	Yes
Jeff Yutzler	Image Matters LLC	Yes
Luís de Sousa	CRP Henri Tudor	Yes
Arnaud Cauchy	Airbus Defence & Space	Yes

# VII

## INTRODUCTION

The WPS standard provides a robust, interoperable, and versatile protocol for process execution on web services. It supports both immediate processing for computational tasks that take little time and asynchronous processing for more complex and time consuming tasks. Moreover, the

WPS standard defines a general process model that is designed to provide an interoperable description of processing functions. It is intended to support process cataloguing and discovery in a distributed environment.

1

# SCOPE

---

This document specifies the interface to a general purpose Web Processing Service (WPS). A WPS is a web service that enables the execution of computing processes and the retrieval of metadata describing their purpose and functionality. Typically, these processes combine raster, vector, and/or coverage data with well-defined algorithms to produce new raster, vector, and/or coverage information.

The WPS protocol supports both synchronous and asynchronous execution of processes. Synchronous execution may be used in simple and quick computation scenarios, where the data processing takes little to almost no time. Asynchronous processing is particularly well suited for complex computation scenarios which may take significant time.

The specification uses a core and extensions model to organize its features:

1. A core conceptual model, defining basic requirements for a web based processing service,
2. A process model to support the description and discovery of processes on the web,
3. A basic data model that supports arbitrary (standard or non-standard) data formats for inputs and outputs,
4. A WPS service model and encoding based on OGC baseline standards, and
5. A Dismiss extension to allow clients to terminate asynchronous processing jobs.



2

# CONFORMANCE

---

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site<sup>1</sup>.

**Table 1** – Conformance classes related to WPS 2.0

CONFORMANCE CLASS	DESCRIPTION	CLAUSE
<a href="http://www.opengis.net/spec/WPS/2.0/conf/service/profile/basic-wps">http://www.opengis.net/spec/WPS/2.0/conf/service/profile/basic-wps</a>	Basic WPS service profile	Annex A.1
<a href="http://www.opengis.net/spec/WPS/2.0/conf/service/synchronous-wps">http://www.opengis.net/spec/WPS/2.0/conf/service/synchronous-wps</a>	Synchronous WPS	
<a href="http://www.opengis.net/spec/WPS/2.0/conf/service/asynchronous-wps">http://www.opengis.net/spec/WPS/2.0/conf/service/asynchronous-wps</a>	Asynchronous WPS	
<a href="http://www.opengis.net/spec/WPS/2.0/conf/process-model-encoding">http://www.opengis.net/spec/WPS/2.0/conf/process-model-encoding</a>	WPS process model encoding	
<a href="http://www.opengis.net/spec/WPS/2.0/conf/service/dismiss-extension">http://www.opengis.net/spec/WPS/2.0/conf/service/dismiss-extension</a>	Dismiss extension	

<sup>1</sup>[www.opengeospatial.org/cite](http://www.opengeospatial.org/cite)



3

# NORMATIVE REFERENCES

---

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Arliss Whiteside Jim Greenwood: OGC 06-121r9, *OGC Web Service Common Implementation Specification*. Open Geospatial Consortium (2010).

Policy SWG: OGC 08-131r3, *The Specification Model – Standard for Modular specifications*. Open Geospatial Consortium (2009).

A. Phillips, M. Davis: IETF RFC 4646, *Tags for Identifying Languages*. RFC Publisher (2006). <https://www.rfc-editor.org/info/rfc4646>.

T. Berners-Lee, R. Fielding, L. Masinter: IETF RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax*. RFC Publisher (2005). <https://www.rfc-editor.org/info/rfc3986>.

ISO: ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*. International Organization for Standardization, Geneva (2004). <https://www.iso.org/standard/40874.html>.

W3C: W3C xmlschema-2, *XML Schema Part 2: Datatypes Second Edition*. World Wide Web Consortium <https://www.w3.org/TR/xmlschema-2/>.



The background features a dark blue field with several thin, light yellow lines intersecting at various points. Three of these intersection points are marked with small yellow dots. One dot is located in the upper right quadrant, another is further to the right, and the third is in the lower left quadrant. The lines create a network of triangles and other geometric shapes across the page.

4

# TERMS AND DEFINITIONS

---

## TERMS AND DEFINITIONS

---

This document uses the terms defined in OGC Policy Directive 49, 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 document and OGC documents do not use the equivalent phrases in the ISO/IEC Directives, Part 2.

This document also uses terms defined in the OGC Standard for Modular specifications (OGC 08-131r3), also known as the ‘ModSpec’. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

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

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r9], 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 standard.

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

### 4.1. dataset

---

collection of data, published or curated by a single agent, and available for access or download in one or more formats

**Note 1 to entry:** The use of ‘collection’ in the definition from [DCAT] is broader than the use of the term collection in this specification.

[SOURCE: W3C vocab-dcat]

### 4.2. distribution

---

represents an accessible form of a **dataset**

Example      a downloadable file, an RSS feed or an API.

[SOURCE: W3C vocab-dcat]

## 4.3. feature

characteristic ALTERNATIVE

character DEPRECATED

---

abstraction of real world phenomena

**Note 1 to entry:** For those unfamiliar with the term ‘feature’, the explanations on [Spatial Things, Features and Geometry](#) in the W3C/OGC Spatial Data on the Web Best Practice document provide more detail.

[**SOURCE:** ISO 19101-1:2014, modified – added alternative term for “characteristic” and deprecated term “character”.]

## 4.4. Process

---

A **process**  $p$  is a function that for each input returns a corresponding output

$$\mu: X \rightarrow Y$$

where  $X$  denotes the domain of arguments  $x$  and  $Y$  denotes the co-domain of values  $y$ . Within this specification, process arguments are referred to as process inputs and result values are referred to as process outputs. Processes that have no process inputs represent value generators that deliver constant or random process outputs.

## 4.5. Process description

---

A process description is an information model that specifies the interface of a process. A process description is used for a machine-readable description of the process itself but also provides some basic information about the process inputs and outputs.



6

# CONVENTIONS

---

This section provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

### 6.1. Abbreviated terms

---

GML	Geography Markup Language
GRS	Coordinate Reference System
HTTP	Hypertext Transfer Protocol
ISO	International Organization for Standardization
KVP	Keyword Value Pair
MIME	Multipurpose Internet Mail Extensions
OGC	Open Geospatial Consortium
UML	Unified Modeling Language
URI	Universal Resource Identifier
URL	Uniform Resource Locator
WPS	Web Processing Service
XML	Extensible Markup Language

### 6.2. Use of the Term “Process”

---

The term process is one of the most used terms both in the information and geosciences domain. If not stated otherwise, this specification uses the term process as an umbrella term for any algorithm, calculation or model that either generates new data or transforms some input data into output data as defined in Clause 4.4.

## 6.3. UML Notation

Unified Modeling Language (UML) static structure diagrams appearing in this specification are used as described in section 5.2 of OGC06-121r9. Further, the following conventions hold:

- UML elements having a package name of “OWS Common” are those defined in the UML model of OWS Common [OGC 06-121r9].
- UML data type Any is used here as an equivalence to XML’s xsd:any.
- UML elements not qualified with a package name are those defined in this standard.

The UML model data dictionary is specified herein in a series of tables. The contents of the columns in these tables are described in section 5.5 of [OGC 06-121r9]. The contents of these data dictionary tables are normative, including any table footnotes.

## 6.4. Namespace Conventions

The following namespaces are used in this document. The prefix abbreviations used constitute conventions used here, but are not normative. The namespaces to which the prefixes refer are normative, however.

PREFIX	NAMESPACE URI	DESCRIPTION
ows	<a href="http://www.opengis.net/ows/2.0">http://www.opengis.net/ows/2.0</a>	OWS Common 2.0 XML Schema
xlink	<a href="http://www.w3.org/1999/xlink">http://www.w3.org/1999/xlink</a>	Definitions for XLINK
xml	<a href="http://www.w3.org/XML/1998/namespace">http://www.w3.org/XML/1998/namespace</a>	XML (required for xml:lang)
xs	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	XML Schema



7

# WPS CONCEPTUAL MODEL

---

## WPS CONCEPTUAL MODEL

The WPS service model defines basic properties of any WPS server. A WPS server is a web service that provides access to pre-defined processes and provides job control operations to instantiate, control and monitor processing jobs (Figure 1).

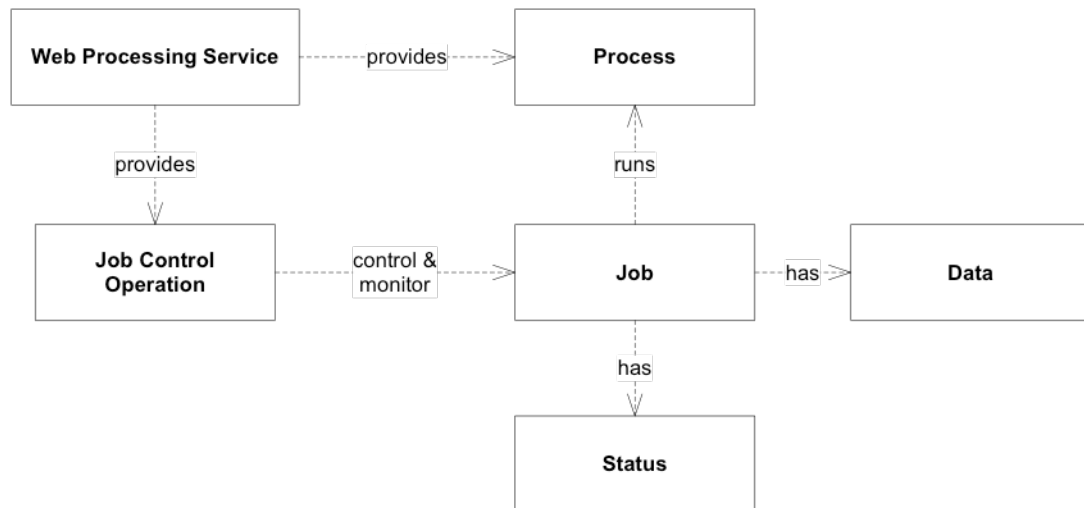


Figure 1 – Artifacts of the WPS service model

### REQUIREMENTS CLASS 1

OBLIGATION	requirement
TARGET TYPE	Derived information model, encoding, and software implementation
NORMATIVE STATEMENTS	Requirement 1-1 Requirement 1-2 Recommendation 1-1 Requirement 1-3 Requirement 1-4 Requirement 1-5 Requirement 1-6

## 7.1. Service Discovery

Any WPS server shall be self-contained, i.e. provide an initial endpoint that can be used by a WPS client to determine the server's capabilities.



## REQUIREMENTS CLASS 2

OBLIGATION	requirement
TARGET TYPE	Derived information model, encoding, and software implementation
NORMATIVE STATEMENTS	Requirement 2-1 Requirement 2-2 Requirement 2-3

## 7.2. Service Capabilities

The basic capabilities of any WPS server fall into two categories: The first category comprises capabilities for process discovery and retrieval of process descriptions. The second category comprises capabilities to manage and monitor processing jobs.

Since the processes provided by a WPS server may have different degrees of complexity, the server shall indicate the allowed job control capabilities mode per process offering.

Further service capabilities, i.e. for secure communication and user authentication may be provided with the service but are neither covered nor restricted by this specification as long as they do not alter or change the semantics of other job control capabilities.

## REQUIREMENTS CLASS 3

OBLIGATION	requirement
TARGET TYPE	Derived information model, encoding, and software implementation
PREREQUISITE	<a href="http://www.opengis.net/spec/WPS/2.0/req/conceptual-model/process">http://www.opengis.net/spec/WPS/2.0/req/conceptual-model/process</a>
NORMATIVE STATEMENTS	Requirement 3-1 Requirement 3-2 Requirement 3-3 Requirement 3-4



8

# WPS NATIVE PROCESS MODEL

---

This section describes the information model of requirements. The corresponding XML and plain text encodings are specified in Clause 9.

## REQUIREMENTS CLASS 4

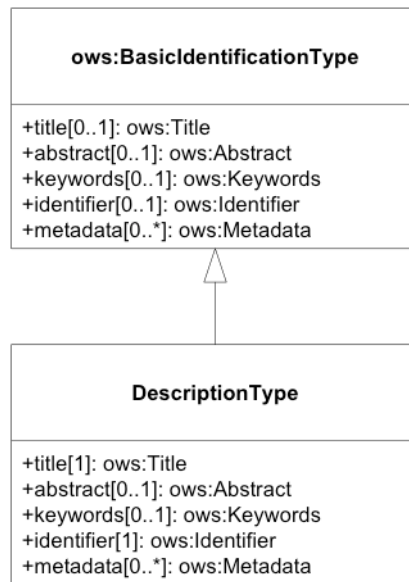
OBLIGATION	requirement
TARGET TYPE	Derived encoding and software implementation
PREREQUISITE	<a href="http://www.opengis.net/spec/WPS/2.0/req/conceptual-model/process">http://www.opengis.net/spec/WPS/2.0/req/conceptual-model/process</a>
NORMATIVE STATEMENTS	Requirement 4-1 Requirement 4-2 Requirement 4-3 Requirement 4-4 Requirement 4-5

## 8.1. Common Description Type

Descriptive elements of processes, inputs and outputs are derived from the BasicIdentificationType provided by OWS Common (Figure 2). Other descriptive information shall be recorded in the Metadata element in the form of simple links with an appropriate role identifier.

## REQUIREMENTS CLASS 5

OBLIGATION	requirement
TARGET TYPE	Derived encoding and software implementation
PREREQUISITES	<a href="http://www.opengis.net/spec/WPS/2.0/req/conceptual-model/process">http://www.opengis.net/spec/WPS/2.0/req/conceptual-model/process</a> OWS Common 2.0 – BasicDescriptionType
NORMATIVE STATEMENTS	Requirement 5-1 Requirement 5-2 Requirement 5-3 Requirement 5-4 Requirement 5-5



**Figure 2** – DescriptionType for processes, process inputs and process outputs UML class diagram

**Table 2** – Properties of the DescriptionType structure

NAMES	DEFINITION	DATA TYPE AND VALUES	MULTIPLICITY AND USE
Title	Title of the process, input, and output. Normally available for display to a human.	ows:Title	One (mandatory)
Abstract	Brief narrative description of a process, input, and output. Normally available for display to a human.	ows:Abstract	Zero or one (optional) Include when available and useful.
Keywords	Keywords that characterize a process, its inputs, and outputs.	ows:Keywords	Zero or more (optional) Include when available and useful.
Identifier	Unambiguous identifier of a process, input, and output.	ows:Identifier Value is a URI or HTTP-URI <sup>a</sup>	One (mandatory)
Metadata	Reference to additional metadata about this item.	ows:Metadata Allowed values are specified in Table 3.	Zero or more (optional)

<sup>a</sup> Additional content such as separate code space and version attributes in the Identifier element are not allowed.

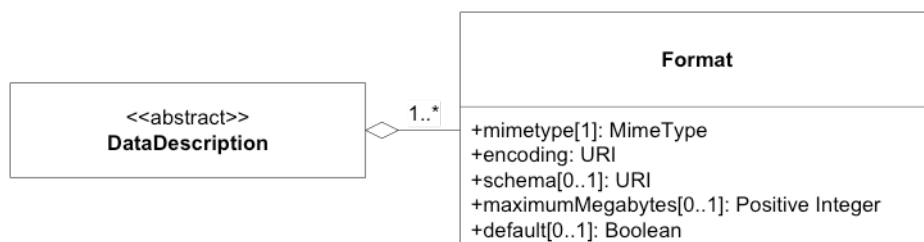
**Table 3** – Properties of the Metadata structure

NAMES	DEFINITION	DATA TYPE AND VALUES	MULTIPLICITY AND USE
Title	Title of the documentation. Normally available for display to a human.	Character String	One (mandatory)
Link type	Type of the xlink, fixed to simple.	Character String, fixed to "simple".	One (mandatory)
Role	Role identifier, indicating the role of the linked document.	HTTP-URI	One (mandatory)
href	Reference to a documentation site for a process, input, or output.	HTTP-URI	One (mandatory)

## 8.2. Data Description Structure

The DataDescription structure contains basic properties for defining data inputs and outputs, including mimetype, encoding and schema. These properties specify supported formats for input and output data of computing processes. Any input or output item may support multiple formats, one of which is the default format. Processes may require that an input or output data set does not exceed a certain data volume.

REQUIREMENTS CLASS 6	
OBLIGATION	requirement
TARGET TYPE	Derived information model, encoding, and software implementation
PREREQUISITE	<a href="http://www.opengis.net/spec/WPS/2.0/req/conceptual-model/process">http://www.opengis.net/spec/WPS/2.0/req/conceptual-model/process</a>
NORMATIVE STATEMENTS	Requirement 6-1 Requirement 6-2



**Figure 3** – DataDescription and supported formats UML class diagram

**Table 4 — Format properties**

NAMES	DEFINITION	DATA TYPE AND VALUES	MULTIPLICITY AND USE
mimetype	Media type of the data.	Character String	One (mandatory)
encoding	Encoding procedure or character set of the data (e.g. raw or base64)	Character String, fixed to "simple".	One (mandatory)
schema	Identification of the data schema.	HTTP-URI	One (mandatory)
maximum Megabytes	The maximum size of the input data, in megabytes.	Integer	Zero or one (optional)
default	Indicates that this format is the default format. <sup>a</sup>	Boolean	Zero or one (conditional) <sup>a, b</sup>

<sup>a</sup> Defaults to FALSE if omitted.

<sup>b</sup> One of the formats included in the DataDescription structure shall have the attribute "default" set to "true".



9

# WPS NATIVE PROCESS MODEL ENCODING

---

## 9.1. XML Schema Implementation

This section specifies the XML encoding of the elements of the WPS native process model. The referred XML schema elements are provided by the associated schema package delivered with this standard and located at <http://schemas.opengis.net/wps/2.0/>.

### REQUIREMENTS CLASS 7

OBLIGATION	requirement
TARGET TYPE	Derived encoding and software implementation
PREREQUISITES	<a href="http://www.opengis.net/spec/WPS/2.0/req/native-process/model">http://www.opengis.net/spec/WPS/2.0/req/native-process/model</a> OWS Common 2.0
NORMATIVE STATEMENTS	Requirement 7-1 Requirement 7-2 Requirement 7-3

### 9.1.1. Data Types

The XML encoding of data types defines encoding rules for ComplexData, LiteralData, BoundingBoxData as well as their values.

Examples for data type encodings are listed in Annex B.1.

### REQUIREMENTS CLASS 8

OBLIGATION	requirement
TARGET TYPE	Derived encoding and software implementation
PREREQUISITES	<a href="http://www.opengis.net/spec/WPS/2.0/req/native-process/model/datatypes/complex-data">http://www.opengis.net/spec/WPS/2.0/req/native-process/model/datatypes/complex-data</a> <a href="http://www.opengis.net/spec/WPS/2.0/req/native-process/model/datatypes/literal-data">http://www.opengis.net/spec/WPS/2.0/req/native-process/model/datatypes/literal-data</a> <a href="http://www.opengis.net/spec/WPS/2.0/req/native-process/model/datatypes/bounding-box-data">http://www.opengis.net/spec/WPS/2.0/req/native-process/model/datatypes/bounding-box-data</a>



## REQUIREMENTS CLASS 8

OWS Common 2.0

### NORMATIVE STATEMENTS

Requirement 8-1  
Requirement 8-2

### 9.1.2. Process Description

This clause specifies the XML encoding for the Process description.

## REQUIREMENTS CLASS 9

### OBLIGATION

requirement

### TARGET TYPE

Derived software implementation

### PREREQUISITES

<http://www.opengis.net/spec/WPS/2.0/req/native-process/model/description>  
OWS Common 2.0

### NORMATIVE STATEMENTS

Requirement 9-1  
Requirement 9-2

### 9.1.3. Generic Process

This clause specifies the XML encoding for the GenericProcess.

## REQUIREMENTS CLASS 10

### OBLIGATION

requirement

### TARGET TYPE

Derived software implementation

### PREREQUISITES

<http://www.opengis.net/spec/WPS/2.0/req/native-process/profile/generic>  
OWS Common 2.0

### NORMATIVE STATEMENTS

Requirement 10-1  
Requirement 10-2

## 9.2. Plain Text Encoding for LiteralData and BoundingBoxData Values

This clause specifies the plain text encoding of data types for literal and bounding box data values.

### REQUIREMENTS CLASS 11

<b>OBLIGATION</b>	requirement
<b>TARGET TYPE</b>	Derived encoding and software implementation
<b>PREREQUISITES</b>	<a href="http://www.opengis.net/spec/WPS/2.0/req/native-process/model/datatypes/complex-data">http://www.opengis.net/spec/WPS/2.0/req/native-process/model/datatypes/complex-data</a> <a href="http://www.opengis.net/spec/WPS/2.0/req/native-process/model/datatypes/literal-data">http://www.opengis.net/spec/WPS/2.0/req/native-process/model/datatypes/literal-data</a> <a href="http://www.opengis.net/spec/WPS/2.0/req/native-process/model/datatypes/bounding-box-data">http://www.opengis.net/spec/WPS/2.0/req/native-process/model/datatypes/bounding-box-data</a> OWS Common 2.0
<b>NORMATIVE STATEMENTS</b>	Requirement 11-1 Requirement 11-2

Literal values - BNF schema:

```
literalvalue = value *1("@datatype=" datatype) *1("@uom=" uom)
value       = 1*VCHAR
datatype    = URI
uom         = URI
```

Literal values - Example:

```
70@datatype=http://www.w3.org/2001/XMLSchema#integer@uom=meter
```

BoundingBox values - BNF schema<sup>2</sup>:

```
bbox      = lc_coords "," uc_coords ["," crs] ["," dimensions]
lc_coords = number ["," number]
uc_coords = number ["," number]
number    = 1*DIGIT["." 1*DIGIT]
crs       = 1*VCHAR
dimensions = 1*DIGIT
```

BoundingBoxData values - Examples:

```
51.9,7.0,53.0,8.0,EPSG:4326
```

<sup>2</sup>The dimensions attribute is included for compliance with the BoundingBox structure defined in OWS Common [OGC 06-121r9]. However, its use should be generally avoided since the number of dimensions is already part of the CRS definition and usually superfluous.

51.9,7.0,53.0,8.0,<http://www.opengis.net/def/crs/EPSG/0/4258>



10

# COMMON WPS SERVICE MODEL

---

A Web Processing Service consists of processes and service operations. By definition, processes represent the computational functionality of a WPS, while service operations are used to interact with the WPS and in particular to use the service's process offerings.

#### REQUIREMENTS CLASS 12

<b>OBLIGATION</b>	requirement
<b>TARGET TYPE</b>	Software implementation
<b>PREREQUISITES</b>	<a href="http://www.opengis.net/spec/WPS/2.0/req/conceptual-model">http://www.opengis.net/spec/WPS/2.0/req/conceptual-model</a> OWS Common 2.0
<b>NORMATIVE STATEMENTS</b>	Requirement 12-1 Requirement 12-2 Requirement 12-3 Requirement 12-4 Requirement 12-5 Requirement 12-6 Requirement 12-7 Requirement 12-8 Requirement 12-9 Requirement 12-10

### 10.1. Overview of WPS Core Operations

The WPS interface specified in this section supports retrieval and execution of processes for geospatial computation. For that purpose, the WPS service model specifies the following operations that may be invoked by a WPS client and performed by a WPS server<sup>3</sup>:

**GetCapabilities** — This operation allows a client to request information about the server's capabilities and processes offered.

**DescribeProcess** — This operation allows a client to request detailed metadata on selected processes offered by a server.

**Execute** — This operation allows a client to execute a process comprised of a process identifier, the desired data inputs and the desired output formats.

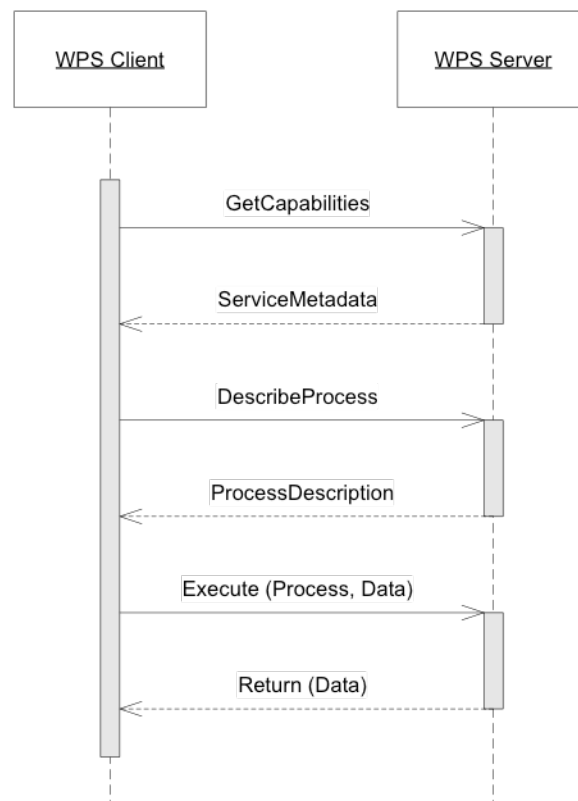
<sup>3</sup>NOTE: Future extensions of this specification may introduce additional operations.

GetStatus — This operation allows a client to query status information of a processing job (conditional).

GetResult — This operation allows a client to query the results of a processing job (conditional).

During a sequence of WPS requests, a client should first issue a GetCapabilities request to the server to obtain an up-to date listing of available processes. Then, it may issue a DescribeProcess request to find out more details about particular processes offered, including the supported data formats. To run a process with the desired input data, a client will issue an Execute request<sup>4</sup> (Figure 4).

The operations GetStatus and GetResult are used in conjunction with asynchronous execution. If a WPS server offers synchronous process execution only, these operations may not be implemented. Detailed guidance is provided by the corresponding profiles and conformance classes.



**Figure 4 — Common sequence of WPS operations UML sequence diagram**

<sup>4</sup>NOTE: A WPS server can change its offering at any time, in particular between a GetCapabilities, a subsequent DescribeProcess, and a subsequent Execute request. Any quality of service (QoS) guarantees are within the responsibilities of the service provider and not covered by this standard.

# 10.2. Data Transmission

Data exchange between WPS clients and servers requires an agreement on the general data exchange patterns and suitable communication protocols. Data may be sent to (received from) a WPS in two distinct ways: (1) by reference (using HTTP/GET or HTTP/POST), and (2) by value. Clients may send input data in either fashion. Output data may be requested in any fashion declared by the data transmission options defined for the process offering. Typically, large data inputs and outputs are delivered by reference.

REQUIREMENTS CLASS 13	
OBLIGATION	requirement
TARGET TYPE	Derived encoding and software implementation
PREREQUISITE	<a href="http://www.opengis.net/spec/WPS/2.0/req/conceptual-model">http://www.opengis.net/spec/WPS/2.0/req/conceptual-model</a>
NORMATIVE STATEMENT	Requirement 13-1

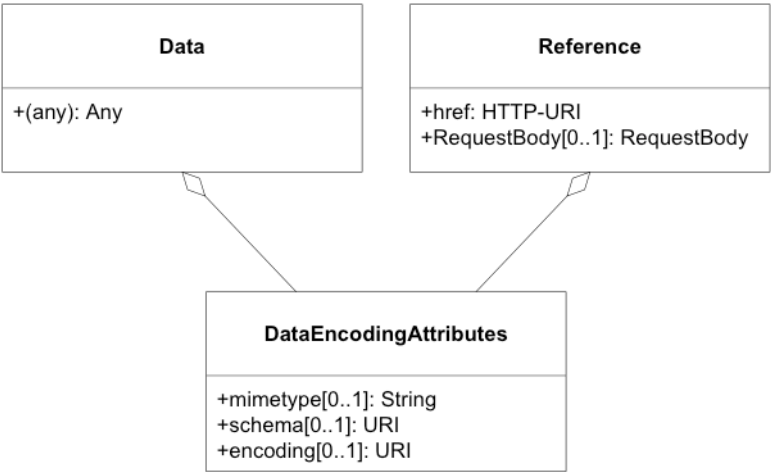


Figure 5 – Input and output data transmission structures UML class diagram

Table 5 – Parts of the inline Data structure

NAMES	DEFINITION	DATA TYPE AND VALUES	MULTIPLICITY AND USE
mimetype	See Table 6 – Properties of the DataEncodingAttributes structure.		
encoding			

NAMES	DEFINITION	DATA TYPE AND VALUES	MULTIPLICITY AND USE
schema			
(any) <sup>a</sup>	The actual input or output data.	Any type and value	One (mandatory)

<sup>a</sup> The data is embedded here as part of the Data element, in the mimeType, encoding, and schema indicated by the first three parameters if they exist, or by the relevant defaults.

**Table 6 — Properties of the DataEncodingAttributes structure**

NAMES	DEFINITION	DATA TYPE AND VALUES	MULTIPLICITY AND USE
mimetype	MimeType of the data.	Character String	One (mandatory)
encoding	Well-known encoding or character set of the data.	URI “raw” shall be used for plain binary data “base64” shall be used for base64 encoded data Character set identifiers (e.g. “UTF-8”) shall be used for text or CSV data.	Zero or one (conditional) <sup>a</sup>
schema	Identification of the data schema.	URI	Zero or one (conditional) <sup>a</sup>

<sup>a</sup> This shall be provided if: 1) the process data item supports multiple encodings / schemas, and 2) the data is not of the default encoding / schema, and 3a) the schema / encoding cannot be retrieved from the data itself, or 3b) the encoding / schema information is deeply buried inside the data (i.e. not part of some header) and requires significant parsing effort.

**Table 7 — Parts of the Reference structure**

NAMES	DEFINITION	DATA TYPE AND VALUES	MULTIPLICITY AND USE
mimetype			
encoding	See Table 6 — Properties of the DataEncodingAttributes structure.		
schema			
href	HTTP URI that points to the remote resource where the data may be retrieved.	HTTP URI	One (mandatory)
Request Body	Request body element that is used for HTTP/ POST requests to the above URL. If no request body is present, an HTTP/GET	Request should be used to retrieve the data. RequestBody structure, see Table 8.	Zero or one (optional)



**Table 8** — Parts of the RequestBody structure

NAMES	DEFINITION	DATA TYPE AND VALUES	MULTIPLICITY AND USE
Body	The contents of this element to be used as the body of the HTTP request message to be sent to the service identified in ../Reference/@href. For example, it could be an XML encoded WFS request using HTTP/POST.	Any type	Zero or one (conditional) <sup>a</sup>
Body Reference	Reference to a remote document to be used as the body of an HTTP/POST request message to the service identified in the href element in the Reference structure (Table 7).	BodyReference, see Table 9.	Zero or one (conditional) <sup>a</sup>

<sup>a</sup> One and only one of these items shall be included.

**Table 9** — Parts of the BodyReference structure

NAMES	DEFINITION	DATA TYPE AND VALUES	MULTIPLICITY AND USE
href	HTTP URI that points to the remote resource where the request body may be retrieved.	HTTP URI	One (mandatory)



# ANNEX A (NORMATIVE) ABSTRACT TEST SUITE

---



# ANNEX A

## (NORMATIVE)

### ABSTRACT TEST SUITE

---

Tests and requirement identifiers below are relative to <http://www.opengis.net/spec/WPS/2.0>

#### A.1. Basic WPS (Conformance Class)

---

The OGC URI identifier of this conformance class is: <http://www.opengis.net/spec/WPS/2.0/conf/service/profile/basic-wps>

Example

##### REQUIREMENT A.1

STATEMENT	conf/service/profile/basic-wps
-----------	--------------------------------

##### RECOMMENDATION A.1

STATEMENT	Verify that the server implements the Basic WPS conformance class.
-----------	--

##### REQUIREMENT A.2

STATEMENT	Verify that the server implements the Synchronous WPS and/or the Asynchronous WPS conformance class. Verify that the requests and responses to a supported operation are syntactically correct. Verify that the service supports the Synchronous WPS Conformance class, the Asynchronous WPS Conformance class or both. Verify that all process offerings implement the native process model.
-----------	---



B

# ANNEX B (INFORMATIVE)

## XML EXAMPLES

---

## B

## ANNEX B (INFORMATIVE) XML EXAMPLES

---

### B.1. Data Types

---

#### B.1.1. Complex Data Description

```
<wps:ComplexData>
  <wps:Format mimeType="application/geotiff" encoding="raw"
default="true"/>
  <wps:Format mimeType="application/geotiff" encoding="base64"/>
</wps:ComplexData>
```

#### B.1.2. Literal Data Description

```
<wps:LiteralData>
  <wps:Format mimeType="text/plain" default="true"/>
  <wps:Format mimeType="text/xml"/>
  <LiteralDataDomain default="true">
    <ows:AllowedValues>
      <ows:Range>
        <ows:MinimumValue>1</ows:MinimumValue>
        <ows:MaximumValue>1000</ows:MaximumValue>
      </ows:Range>
    </ows:AllowedValues>
    <ows:DataType
ows:reference="http://www.w3.org/2001/XMLSchema#float">float
</ows:DataType>
    <ows:UOM>meters</ows:UOM>
    <ows:DefaultValue>100</ows:DefaultValue>
  </LiteralDataDomain>
  <LiteralDataDomain>
    <ows:AllowedValues>
      <ows:Range>
        <ows:MinimumValue>1</ows:MinimumValue>
        <ows:MaximumValue>3000</ows:MaximumValue>
      </ows:Range>
    </ows:AllowedValues>
    <ows:DataType
ows:reference="http://www.w3.org/2001/XMLSchema#float">float
</ows:DataType>
    <ows:UOM>feet</ows:UOM>
  </LiteralDataDomain>
```

```
</wps:LiteralData>
```

### B.1.3. Literal data values

```
<LiteralValue  
  dataType=http://www.w3.org/2001/XMLSchema#double  
  uom="meter">  
    42.1  
</LiteralValue>
```

```
<LiteralValue  
  dataType="http://www.w3.org/2001/XMLSchema#string">  
    ArableLand  
</LiteralValue>
```

### B.1.4. BoundingBox Data Description

```
<wps:BoundingBoxData>  
  <wps:Format mimeType="text/plain" default="true"/>  
  <wps:Format mimeType="text/xml"/>  
  <wps:SupportedCRS default="true">EPSG:4326</wps:SupportedCRS>  
  <wps:SupportedCRS>  
    http://www.opengis.net/def/crs/EPSG/0/4258  
  </wps:SupportedCRS>  
</wps:BoundingBoxData>
```

### B.1.5. BoundingBox Data Values

```
<ows:BoundingBox crs="EPSG:4326">  
  <ows:LowerCorner>51.9 7.0</ows:LowerCorner>  
  <ows:UpperCorner>53.0 8.0</ows:UpperCorner>  
</ows:BoundingBox>  
  
<ows:BoundingBox  
  crs="http://www.opengis.net/def/crs/EPSG/0/4258">  
  <ows:LowerCorner>51.9 7.0</ows:LowerCorner>  
  <ows:UpperCorner>53.0 8.0</ows:UpperCorner>  
</ows:BoundingBox>
```



# BIBLIOGRAPHY





## BIBLIOGRAPHY

---

- [1] ISO: ISO 19101, *Geographic information – Reference model*. International Organization for Standardization, Geneva <https://www.iso.org/standard/26002.html>.
- [2] W3C: W3C vocab-dcat, *Data Catalog Vocabulary (DCAT)*. World Wide Web Consortium <https://www.w3.org/TR/vocab-dcat/>.