

OGC Testbed-14
WPS-T Engineering Report

Table of Contents

1. Summary	4
1.1. Requirements & Research Motivation	4
1.2. Prior-After Comparison	4
1.3. Recommendations for Future Work	4
1.4. Document contributor contact points	4
1.5. Foreword	5
2. References	6
3. Terms and definitions	7
3.1. Abbreviated terms	7
4. Overview	8
5. GetCapabilities Operation Update	9
5.1. Introduction	9
5.2. GetCapabilities Operation Response Update	9
5.2.1. Normal response update	9
5.2.2. OperationsMetadata section standard contents	9
5.2.3. Constraint element for DeployProcess operation	10
5.2.4. Capabilities document example	11
6. DeployProcess Operation (Mandatory)	13
6.1. Introduction	13
6.2. Transactional and atomic behavior of DeployProcess operation	13
6.3. DeployProcess operation request	13
6.3.1. DeployProcess request parameters	13
6.3.2. DeployProcess request KVP encoding	15
6.3.3. DeployProcess request XML encoding	16
6.4. DeployProcess operation response	17
6.4.1. Normal response parameters	17
6.4.2. DeployProcess response example	17
6.4.3. DeployProcess exceptions	18
7. UndeployProcess Operation (mandatory)	19
7.1. Introduction	19
7.2. Transactional and atomic behavior of DeployProcess operation	19
7.3. UndeployProcess operation request	19
7.3.1. UndeployProcess request parameters	20
7.3.2. UndeployProcess request KVP encoding	20
7.3.3. UndeployProcess request XML encoding	21
7.4. UndeployProcess operation response	21
7.4.1. Normal response parameters	22
7.4.2. UndeployProcess response example	22

7.4.3. UndeployProcess exceptions	22
8. BPMN Deployment Profile	24
8.1. Introduction.....	24
Appendix A: Abstract Test Suite	26
Appendix B: XML Schema Documents	27
Appendix C: Revision History	36
Appendix D: Bibliography	37

Publication Date: 2019-02-07

Approval Date: 2018-12-13

Submission Date: 2018-11-20

Reference number of this document: OGC 18-036r1

Reference URL for this document: <http://www.opengis.net/doc/PER/t14-D025>

Category: Public Engineering Report

Editor: Benjamin Pross, Arnaud Cauchy

Title: OGC Testbed-14: WPS-T Engineering Report

OGC Engineering Report

COPYRIGHT

Copyright (c) 2019 Open Geospatial Consortium. To obtain additional rights of use, visit <http://www.opengeospatial.org/>

WARNING

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

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.

Chapter 1. Summary

This Engineering Report describes a proposed transactional extension for the OGC Web Processing Service (WPS) 2.0 standard including Key-Value Pair (KVP) and Extensible Markup Language (XML) bindings and recommendations for a process deployment profile for BPMN (Business Process Model and Notation).

1.1. Requirements & Research Motivation

The need to dynamically add custom processing functionality to servers is steadily growing. For example, in the Earth Observation community the custom is to move the processing functionality to the data, typically using containers. Also, the creation of workflows and the dynamic upload of them is an area of growing interest.

This ER builds on lessons from the previous testbed, Testbed-13, which were documented and presented by Pross and Stasch [1].

1.2. Prior-After Comparison

Currently, no transactional extension is available for WPS 2.0. An unofficial discussion paper is currently available that is the basis for this report. That discussion paper uses concepts from WPS 1.0. This Engineering Report defines a transactional extension for WPS 2.0. In abbreviated form, the transactional extension of WPS is referred to as WPS-T. The ER also provides the schemas and abstract test suite for the extension.

1.3. Recommendations for Future Work

This document should be the foundation for a transactional extension for the WPS 2.0 standard. The following aspects need further investigation:

- Security
- Discovery of deployment profiles
- WPS-T Representational State Transfer (REST) binding

1.4. Document contributor contact points

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

Contacts

Name	Organization
Benjamin Pross	52°North
Arnaud Cauchy	Airbus
Christophe Noel	Spacebel
Mark Lawrence	Compusult

1.5. Foreword

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.

Chapter 2. References

- [OGC: OGC 14-064r2r9, OGC® WPS 2.0.2 Interface Standard, 2010](http://docs.opengeospatial.org/is/14-065/14-065.html) [http://docs.opengeospatial.org/is/14-065/14-065.html]
- [OGC: OGC 06-121r9, OGC® Web Services Common Standard, 2015](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2]

Chapter 3. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2] shall apply. In addition, the following terms and definitions apply.

- Container

A standardized unit of software

- Workflow

Automation of a process, in whole or part, during which electronic documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules (source ISO 12651-2:2014)

3.1. Abbreviated terms

- BPMN Business Process Model and Notation
- COM Component Object Model
- COTS Commercial Off The Shelf
- WPS Web Processing Service
- WPS-T Transactional Web Processing Service

Chapter 4. Overview

Section 5 introduces an update to the WPS 2.0 GetCapabilities operation.

Section 6 describes the DeployProcess operation.

Section 7 presents the UndeployProcess operation.

Section 8 provides an overview of the BPMN DeploymentProfile specified in Testbed-14.

Chapter 5. GetCapabilities Operation Update

This section describes the update to the GetCapabilities operation.

5.1. Introduction

The WPS 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 the offered processes. The extension of the Web Processing Service 2.0 interface with two new operations requires the modification of the GetCapabilities Metadata accordingly:

- Operation Metadata element shall be updated with two new Operation names "DeployProcess" and "UndeployProcess".
- For the specific case of DeployProcess Operation Metadata, the Constraint element is used to specify a list of supported process deployment profiles and consequently shall become mandatory.

This clause specifies an additional part of the XML document that a WPS 2.0 server implementing WPS-T extension shall return to describe its transactional capabilities.

5.2. GetCapabilities Operation Response Update

5.2.1. Normal response update

In addition to sections specified in [OGC 14-065] - Web Processing Service 2.0 that specify common parts of an RPC-style implementation, the service metadata document shall contain revised sections specified in [Table 1](#)

Table 1. Section name values and contents

Section name	Contents
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 OWSSs, as specified in Subclause 7.4.6 and owsOperationsMetadata.xsd of [OGC 06-121r9] OGC Web Service Common Implementation Specification 2.0 and comply with specificities of [OGC 14-065r2] Web Processing Service 2.0 Interface Standard.

5.2.2. OperationsMetadata section standard contents

For the WPS-T, 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 05-008] plus WPS 2.0 specificities specified in [OGC 14-065] WPS 2.0 Interface Standard. The mandatory values of various (XML) attributes shall be as specified in [Table 2](#). Similarly, the optional attribute values listed in [Table 3](#) shall be included, or not, depending on whether that operation is implemented by that

server. In [Table 2](#) and [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 2. 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.
	GetStatus	The GetStatus operation is implemented by this server.
	GetResult	The GetResult operation is implemented by this server.

Table 3. Optional values of OperationsMetadata section attributes

Attribute name	Attribute value	Meaning of attribute value
Operation.name	DeployProcess	The DeployProcess operation is implemented by this server.
	UndeployProcess	The UndeployProcess operation is implemented by this server.

5.2.3. Constraint element for DeployProcess operation

Operation element for the operation DeployProcess uses the included Constraint element to specify the list of supported process deployment profiles. This Constraint element includes the element listed in [Table 4](#).

Table 4. Parameters included in Constraint section

Names	Definition	Data type and values	Multiplicity and use
SupportedDeploymentProfiles	List of supported process deployment profiles - see Table 5	Ows:DomainType	Zero on One (conditional [1: Mandatory for Operation "DeployProcess"])

The SupportedDeploymentProfiles elements include elements listed in [Table 5](#)

Table 5. Parts of SupportedDeploymentProfiles data structure

Names	Definition	Data type and values	Multiplicity and use
Default DeploymentSchema	Default deployment profile schema [2: Default deployment schema shall exist in the list of SupportedDeploymentSchema.] - see Table 6	DeploymentSchema	One (Mandatory)
Supported DeploymentSchema	List of deployment profile schema - see Table 6	DeploymentSchema	One or more (mandatory)

The DeploymentSchema elements includes element listed in [Table 6](#)

Table 6. Parts of DeploymentSchema data structure

Names	Definition	Data type and values	Multiplicity and use
SchemaName	Name of the deployment schema	string	One (Mandatory)
SchemaReference	xlink to the deployment profile schema	xlink:href	Zero to One (Optional)

5.2.4. Capabilities document example

In response to a GetCapabilities operation request, a WPS 2.0 server which implement the WPS-T extension might generate a document for the section OperationMetadata that looks like:

```

<ows:Operation name="DeployProcess">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://yourhost/wps"/>
    </ows:HTTP>
  </ows:DCP>
  <ows:Constraint>
    <wps:SupportedDeploymentProfiles>
      <wps:Default>
        <wps:DeploymentSchema
          name="BPMN"
          xlink:href="https://www.omg.org/spec/BPMN/20100501/BPMN20.xsd" />
        </wps:Default>
      <wps:Supported>
        <wps:DeploymentSchema
          name="BPMN"
          xlink:href="https://www.omg.org/spec/BPMN/20100501/BPMN20.xsd" />
        <wps:DeploymentSchema
          name="JavaVM"/>
      </wps:Supported>
    </wps:SupportedDeploymentProfiles>
  </ows:Constraint>
</ows:Operation>
<ows:Operation name="UndeployProcess">
  <ows:DCP>
    <ows:HTTP>
      <ows:Post xlink:href="http://yourhost/wps"/>
    </ows:HTTP>
  </ows:DCP>
</ows:Operation>

```

Chapter 6. DeployProcess Operation (Mandatory)

In the following sections, the DeployProcess operation is specified.

6.1. Introduction

The DeployProcess operation allows WPS 2.0 clients to dynamically deploy a process into a WPS 2.0 server which implements the WPS 2.0 Transactional Extension. On successful deployment, this process is now part of the WPS 2.0 server processing offering. Consequently, a WPS 2.0 client will be able to query its detailed description and request an execution as specified in [OGC 14-065r2] WPS 2.0 Standard Interface.

6.2. Transactional and atomic behavior of DeployProcess operation

The DeployProcess operation impacts the result and/or behavior of:

- GetCapabilities operation
- DescribeProcess operation
- Execute operation

DeployProcess shall have an atomic behavior:

- On successful deployment, the process shall be part of the process offerings.
- On failed deployment, the process shall not be part of the process offerings in its full meaning.

From transactional point of view, process offering shall be impacted after the successful completion of the DeployProcess operation only.

6.3. DeployProcess operation request

In the following sections, the DeployProcess operation request is specified.

6.3.1. DeployProcess request parameters

A request to perform the DeployProcess operation shall include the use of the data structure specified in [Table 7](#). This table also specifies the Unified Modeling Language (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 “Names” column appear to contain spaces, they shall not contain spaces.

NOTE

To reduce the need for readers to refer to other documents, the first three parameters listed below are largely copied from Table 29 in Subclause 9.2.1 of [OGC 06-121r9].

Table 7. Parameters in DeployProcess operation request

Name [3: The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r9]]	Definition	Data type and values	Multiplicity and use
Service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, fixed to "WPS"	One (Mandatory)
Request	Operation name	Character String type, not empty Value is operation name, fixed to "DeployProcess"	One (mandatory)
Version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
ProcessDescription	Full description of process, including all input and output parameters	ProcessDescriptionChoice data structure - see Table 8	Zero or One (optional) [4: This may be generated from the ExecutionUnit.]
ExecutionUnit	Concrete executable unit of the process	ExecutionUnit data structure - see Table 9	One or more (mandatory)
DeploymentProfileName	Name of the process profile deployment	Character String type, not empty	Zero or One (optional) [5: If not specified, default deployment profile would be used.]
immediateDeployment	Deploy the ExecutionUnit on the first execution	Boolean, default value is true.	Zero to One (optional)

Table 8. Parts of ProcessDescriptionChoice data structure

Name	Definition	Data type and values	Multiplicity and use
Reference	Identifies process description as a web accessible resource, and references that resource	Reference data structure, see [OGC 14-065r2], Table 25	Zero or one (conditional) [6: One and only one of these two items shall be included.]

Name	Definition	Data type and values	Multiplicity and use
ProcessOffering	Identifies process description as being encapsulated in the DeployProcess request	ProcessOffering data structure - see [OGC 14-065r2], Table 40	Zero or one (conditional) [6: One and only one of these two items shall be included.]

Note: It may be useful to include additional metadata in the ProcessOffering.

Table 9. Parts of ExecutionUnit data structure

Name	Definition	Data type and values	Multiplicity and use
Reference	Identifies the ExecutionUnit as a web accessible resource, and references that resource	Reference data structure, see [OGC 14-065r2], Table 25	Zero or one (conditional) [7: One and only one of these two items shall be included.]
Unit	Identifies the ExecutionUnit being encapsulated in the DeployProcess request	Any	Zero or one (conditional) [7: One and only one of these two items shall be included.]

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 not be empty.

The “Multiplicity and use” columns in [Table 7](#) through [Table 9](#) specify the optionality of each listed parameter and data structure in the DeployProcess operation request. All the “mandatory” parameters and data structures shall be implemented by all WPS 2.0 clients, using a specified value(s). Similarly, all the “mandatory” parameters and data structures shall be implemented by all WPS 2.0 servers, checking that each request parameter or data structure is received with any specified value(s). All the “optional” parameters and data structures, in the DeployProcess operation request, should be implemented by all WPS 2.0 clients using specified values. Similarly, all the “optional” parameters and data structures shall be implemented by all WPS 2.0 servers.

6.3.2. DeployProcess request KVP encoding

Servers may implement HTTP GET transfer of the DeployProcess operation request, using KVP encoding. The KVP encoding of the DeployProcess operation request shall use the parameters specified in [Table 10](#). The parameters listed in [Table 10](#) shall be as specified in [Table 7](#) above.

Table 10. DeployProcess operation request URL parameters

Name and example [8: 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 06-121r9]]	Optionality and use	Definition and format
service=WPS	Mandatory	Service type identifier
request=DeployProcess	Mandatory	Operation name
version=2.0.0	Mandatory	Standard and schema version for this operation
processdescription=http://foo.bar/BufferProcessDescription.xml	Optional	Reference to a full process description
executionunit= http://foo.bar/BufferProcessExecutable.war	Mandatory	Reference to a concrete executable unit of the process
deploymentprofilename="ApplicationServer"	Optional	Name of the process profile deployment
immediatedeployment=true	Optional	Boolean

EXAMPLE

An example DeployProcess operation request KVP encoded for HTTP GET is:

<http://foo.bar/foo?service=WPS&request=DeployProcess&version=2.0.0&processdescription=http://foo.bar/BufferProcessDescription.xml&executionunit=http://foo.bar/BufferProcessExecutable.war&deploymentprofilename=ApplicationServer&immediatedeployment=true>

6.3.3. DeployProcess request XML encoding

All WPS 2.0 servers which support the WPS-T extension should implement HTTP POST transfer of the DeployProcess operation request, using XML encoding.

EXAMPLE

An example DeployProcess operation request, encoded in XML for HTTP POST, is:

```

----
<?xml version="1.0" encoding="UTF-8"?>
<wps:DeployProcess xmlns:wps="http://www.opengis.net/wps/2.0"
  xmlns:ows="http://www.opengis.net/ows/2.0" xmlns:xlink=
"http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/2.0 ../wps.xsd" service="WPS"
  version="2.0.0">
  <wps:ProcessDescription>
    <wps:Reference xlink:href="http://foo.bar/BufferProcessDescription.xml" />
  </wps:ProcessDescription>
  <wps:ExecutionUnit>
    <wps:Reference xlink:href="http://foo.bar/BufferProcessExecutable.war" />
  </wps:ExecutionUnit>
  <wps:DeploymentProfileName>ApplicationServer</wps:DeploymentProfileName>
</wps:DeployProcess>
----

```

6.4. DeployProcess operation response

In the following sections, the DeployProcess operation response is specified.

6.4.1. Normal response parameters

The normal response to a valid DeployProcess operation request shall be a DeploymentResult document. WPS 2.0 server may have updated the process identifier in case of collision with an existing process. WPS 2.0 client shall check the returned process identifier. More precisely, a response from the DeployProcess operation shall include the parts listed in [Table 11](#). This table also specifies the UML model data type plus the multiplicity and use of each listed part.

Table 11. Parts of DeployProcess operation response

Names	Definition	Data type and values	Multiplicity and use
ProcessSummary	Brief information about the process	ProcessSummary data structure - see [OGC 14-065r2], Table 37	One(mandatory)

6.4.2. DeployProcess response example

A DeployProcess operation response can look like this, encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:DeploymentResult service="WPS" version="2.0.0" xsi:schemaLocation=
"http://www.opengis.net/wps/2.0 http://schemas.opengis.net/wps/2.0/wps.xsd" xmlns:wps
="http://www.opengis.net/wps/2.0" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ows=
"http://www.opengis.net/ows/2.0">
  <wps:ProcessSummary>
    <ows:Title>Compute Buffer</ows:Title>
    <ows:Identifier>BufferProcess</ows:Identifier>
  </wps:ProcessSummary/>
</wps:DeploymentResult/>
```

6.4.3. DeployProcess exceptions

When a WPS 2.0 server implementing WPS-T encounters an error while performing a DeployProcess operation, it shall return an exception report message as specified in Subclause 8 of [OGC 06-121r9]. 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 12.

NOTE To reduce the need for readers to refer to other documents, the first five values listed below are copied from Table 28 in Subclause 8.6 of [OGC 06-121r9].

Table 12. Parts of DeployProcess operation response

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	Name of parameter with invalid value
OptionNotSupported	Request is for an option that is not supported by this server	Identifier of option not supported
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit “locator” parameter
DeploymentProfileNotSupported	Deployment of the process required a deployment profile which is not supported by the server	Process identifier not deployed.

Chapter 7. UndeployProcess Operation (mandatory)

In the following sections, the UndeployProcess operation is specified.

7.1. Introduction

The UndeployProcess operation allows WPS 2.0 clients to dynamically undeploy a process from a WPS 2.0 server which implements the WPS 2.0 Transactional Extension. The UndeployProcess operation is applicable to processes that have been previously deployed dynamically. On successful undeployment, the undeployed process is no more part of the process offering of the WPS 2.0 server with all the consequence. Especially a request for execution shall not be accepted. A successful UndeployProcess operation shall result in the permanent removal of the associated execution unit unless parameters of the UndeployProcess operation explicitly request the server to keep the execution unit. In any case, all associated data shall be permanently deleted.

7.2. Transactional and atomic behavior of DeployProcess operation

The UndeployProcess operation impacts the result and/or behavior of the following WPS 2.0 operations:

- GetCapabilities operation
- DescribeProcess operation
- Execute operation

The UndeployProcess shall have an atomic behavior:

- On successful undeployment, the process is no more part of the process offerings.
- On failed undeployment, the process shall remain part of the process offerings in its full meaning.

From a transactional point of view, as soon as an UndeployProcess request is received and accepted by a WPS 2.0 server, the associated process shall no more be part of the process offering. More specifically, the process offering is updated before running into the undeployment of the process. Consequently, later requests shall have a result or a behavior that reflects the process is no more part of the process offering. It is left up to the implementer to decide which behavior the WPS 2.0 server should have with Jobs (and its associated output results) of the involved Process currently managed by the WPS 2.0 server when UndeployProcess operation is received.

7.3. UndeployProcess operation request

In the following sections, the UndeployProcess operation request is specified.

7.3.1. UndeployProcess request parameters

A request to perform the UndeployProcess operation shall include the use of the data structure specified in Table 13. 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 “Names” column appear to contain spaces, they shall not contain spaces.

Table 13. Parameters in UndeployProcess operation request

Name [9: The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 06-121r9]]	Definition	Data type and values	Multiplicity and use
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, fixed to "WPS"	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	Unambiguous identifier or name of a process	Process Identifier - see [OGG 14-065r2], Table 4	One (mandatory)
keepExecutionUnit	Do not remove process execution unit to speed up future deployment	Boolean, default value is false.	Zero to One (optional)

7.3.2. UndeployProcess request KVP encoding

Servers may implement HTTP GET transfer of the UndeployProcess operation request, using KVP encoding. The KVP encoding of the UndeployProcess operation request should use the parameters specified in Table 14. The parameters listed in Table 14 shall be as specified in Table 13 above.

Table 14. UndeployProcess operation request URL parameters

Name and example [10: 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 06-121r9]]	Optionality and use	Definition and format
service=WPS	Mandatory	Service type identifier
request=UndeployProcess	Mandatory	Operation name
version=2.0.0	Mandatory	Standard and schema version for this operation
Identifier=Buffer	Mandatory	Process identifier
keepExecutionUnit=false	Optional	Boolean

EXAMPLE

An example UndeployProcess operation request, encoded as KVP for HTTP GET, is:

<http://foo.bar/foo?service=WPS&request=UndeployProcess&version=2.0.0&identifier=Buffer&keepExecutionUnit=false>

7.3.3. UndeployProcess request XML encoding

All WPS 2.0 servers which support the WPS-T extension should implement HTTP POST transfer of the UndeployProcess operation request, using XML encoding.

EXAMPLE

An example UndeployProcess operation request, encoded in XML for HTTP POST, is:

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:UndeployProcess xmlns:wps="http://www.opengis.net/wps/2.0" xmlns:ows=
"http://www.opengis.net/ows/2.0" xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi
="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"http://www.opengis.net/wps/2.0/wps.xsd" service="WPS" version="2.0.0">
  <ows:Identifier>Buffer</ows:Identifier>
</wps:UndeployProcess>
```

7.4. UndeployProcess operation response

In the following sections, the UndeployProcess operation response is specified.

7.4.1. Normal response parameters

The normal response to a valid UndeployProcess operation request shall be an UndeploymentResult document. More precisely, a response from the UndeployProcess operation shall include the parts listed in Table 15. This table also specifies the UML model data type plus the multiplicity and use of each listed part.

Table 15. Parts of UndeployProcess operation response

Name	Definition	Data type and values	Multiplicity and use
Identifier	Unambiguous identifier or name of a process	Process Identifier - see [OGG 14-065r2], Table 4	One (mandatory)

7.4.2. UndeployProcess response example

An UndeployProcess operation response can look like this encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<wps:UndeploymentResult service="WPS" version="2.0.0" xsi:schemaLocation=
"http://www.opengis.net/wps/2.0 http://schemas.opengis.net/wps/2.0.0/wps.xsd"
xmlns:wps="http://www.opengis.net/wps/2.0" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:ows=
"http://www.opengis.net/ows/2.0">
  <ows:Identifier>Buffer</ows:Identifier>
</wps:UndeploymentResult/>
```

7.4.3. UndeployProcess exceptions

When a WPS 2.0 server implementing WPS-T encounters an error while performing a UndeployProcess operation, it shall return an exception report message as specified in Subclause 8 of [OGC 06-121r9]. The allowed standard exception codes shall include those listed in Table 16. For each listed exceptionCode, the contents of the “locator” parameter value shall be as specified in the right column of Table 16.

Table 16. Parts of UndeployProcess operation response

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	Name of parameter with invalid value
OptionNotSupported	Request is for an option that is not supported by this server	Identifier of option not supported

exceptionCode	value	Meaning of code "locator" value
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit "locator" parameter
UndeploymentDenied	Process not dynamically deployed	Process identifier.

Chapter 8. BPMN Deployment Profile

In the following sections, a BPMN Deployment Profile is specified.

8.1. Introduction

Processes can be deployed in many ways, e.g. by specifying the URL of a docker container or uploading jar-files or scripts to the WPS. It is therefore necessary to specify profiles that describe the mechanisms and data formats of the deployment.

In Testbed-14, the WPS-T works together with a BPMN-Engine. The mechanisms are described in detail in the OGC Testbed-14: BPMN Workflow Engineering Report [OGC 18-085].

For sending the BPMN document to the WPS-T, the ProcessOffering-element is used. The element allows, as subelements, either a Process-element or any other element from a different namespace. So using BPMN is conformant. An example is shown in the following (BPMN largely omitted for better readability):

```

<?xml version="1.0" encoding="UTF-8"?>
<wps:DeployProcess version="2.0.0" service="WPS"
  xmlns:wps="http://www.opengis.net/wps/2.0" xmlns:ows=
"http://www.opengis.net/ows/2.0"
  xmlns:owc="http://www.opengis.net/owc/1.0" xmlns:eoc=
"http://www.opengis.net/wps/2.0/profile/tb13/eoc"
  xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wps/2.0 ../wps.xsd">
  <wps:ProcessOffering jobControlOptions="async-execute dismiss" outputTransmission
="value reference">
    <bpmn2:definitions xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:bpmn2="http://www.omg.org/spec/BPMN/20100524/MODEL"
      xmlns:bpmndi="http://www.omg.org/spec/BPMN/20100524/DI" xmlns:dc=
"http://www.omg.org/spec/DD/20100524/DC"
      xmlns:di="http://www.omg.org/spec/DD/20100524/DI" xmlns:ext=
"http://org.eclipse.bpmn2/ext"
      xmlns:xs="http://www.w3.org/2001/XMLSchema" id="Definitions_1"
      targetNamespace="http://org.eclipse.bpmn2/default/process" exporter=
"org.eclipse.bpmn2.modeler.core"
      exporterVersion="1.1.2.Final">
      <bpmn2:itemDefinition id="ItemDefinition_1"
        structureRef="xs:string" />
      ....
    </bpmn2:definitions>
  </wps:ProcessOffering>
  <wps:ExecutionUnit>
    <wps:Reference href="http://ogc14.compusult.net/wes/wpst/bpmnExecutor" />
  </wps:ExecutionUnit>
  <wps:DeploymentProfileName>BPMN</wps:DeploymentProfileName>
</wps:DeployProcess>

```

The DeploymentProfileName corresponds to a SupportedDeploymentProfile advertized in the capabilities document of the WPS-T.

Appendix A: Abstract Test Suite

Table 17. A.1.1 Capabilities

Test identifier	getcapabilities
Test purpose:	Confirm that the WPS-T advertises the transactional functionality.
Test method:	Request the capabilities. Verify that a SupportedDeploymentProfiles is present and that the DeployProcess and UndeployProcess operations are listed in the OperationsMetadata section.
Requirement:	Section GetCapabilities Operation Update
Test type:	Capability

Table 18. A.1.2 Test case for DeployProcess

Test identifier	deployprocess
Test purpose:	The DeployProcess operation works as expected.
Test method:	Send a valid DeployProcess request to the WPS-T server. Verify that a valid DeploymentResult document is returned.
Requirement:	Section DeployProcess Operation (Mandatory)
Test type:	Capability

Table 19. A.1.3 Test case for UndeployProcess

Test identifier	undeployprocess
Test purpose:	The UndeployProcess operation works as expected.
Test method:	Send a valid UndeployProcess request to the WPS-T server. Verify that a valid UndeploymentResult document is returned.
Requirement:	Section UndeployProcess Operation (mandatory)
Test type:	Capability

Appendix B: XML Schema Documents

The term “XML schema“ means all the XML schema parts having the same XML namespace, usually separated into multiple XML Schema Document files (with the file type “.xsd“. The XML schema parts in one XML namespace are usually separated into multiple XML Schema Documents to ease human understanding.

In addition to this document, this report includes several XML Schema Documents. These XML Schema Documents can be found in the following GitHub repository: <https://github.com/spacebel/common-xml/tree/spacebel/tb14/wps-t/52n-ogc-schema/src/main/resources/META-INF/xml/wps/t>

They will be moved to the official OGC schema repository after the WPS-T extension described in this document will be published.

The capabilities response schema was modified to allow the specification of the supported deployment profiles.

wpsGetCapabilities.xsd

```
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ows=
"http://www.opengis.net/ows/2.0" xmlns:wps="http://www.opengis.net/wps/2.0"
targetNamespace="http://www.opengis.net/wps/2.0" elementFormDefault="qualified"
version="2.0.0" xml:lang="en">
  <annotation>
    <documentation>
      WPS is an OGC Standard.
      Copyright (c) 2015 Open Geospatial Consortium.
      To obtain additional rights of use, visit
      http://www.opengeospatial.org/legal/.
    </documentation>
  </annotation>
  <!-- all-components document include (OGC 06-135r11 s#14) -->
  <include schemaLocation="wps.xsd"/>
  <!-- === IMPORTS === -->
  <import namespace="http://www.opengis.net/ows/2.0" schemaLocation=
"http://schemas.opengis.net/ows/2.0/owsAll.xsd"/>
  <!-- === INCLUDES === -->
  <include schemaLocation="supportedDeploymentProfiles.xsd"/>
  <!-- ===== -->
  <!-- GetCapabilities request elements and types -->
  <!-- ===== -->
  <element name="GetCapabilities" type="wps:GetCapabilitiesType">
    <annotation>
      <documentation>Request to a WPS server to perform the GetCapabilities
operation. This operation allows a client to retrieve a Capabilities XML document
providing metadata for the specific WPS server. </documentation>
    </annotation>
  </element>
  <complexType name="GetCapabilitiesType">
    <complexContent>
```

```

    <extension base="ows:GetCapabilitiesType">
      <attribute name="service" type="ows:ServiceType" use="required" fixed
="WPS">
        <annotation>
          <documentation>service type identifier</documentation>
        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>
<!-- ===== -->
<!-- ===== -->
<!-- GetCapabilities response elements and types -->
<!-- ===== -->
<element name="Capabilities" type="wps:WPSCapabilitiesType">
  <annotation>
    <documentation>WPS GetCapabilities operation response. This document
provides clients with service metadata about a specific service instance, including
metadata about the processes that can be executed. Since the server does not implement
the updateSequence and Sections parameters, the server shall always return the
complete Capabilities document, without the updateSequence parameter. </documentation>
  </annotation>
</element>
<!-- ===== -->
<complexType name="WPSCapabilitiesType">
  <complexContent>
    <extension base="ows:CapabilitiesBaseType">
      <sequence>
        <element ref="wps:Contents"/>
        <element ref="wps:SupportedDeploymentProfiles" minOccurs="0"/>
        <element name="Extension" minOccurs="0">
          <annotation>
            <documentation>container for elements defined by extension
specifications</documentation>
          </annotation>
        </complexType>
        <sequence>
          <any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        </complexType>
      </element>
    </sequence>
    <attribute name="service" use="required" fixed="WPS"/>
  </extension>
</complexContent>
</complexType>
<!-- ===== -->
<element name="Contents">
  <annotation>
    <documentation>List of brief descriptions of the processes offered by this

```

```

WPS server. </documentation>
  </annotation>
  <complexType>
    <sequence>
      <element name="ProcessSummary" type="wps:ProcessSummaryType"
maxOccurs="unbounded">
        <annotation>
          <documentation>Unordered list of one or more brief
descriptions of all the processes offered by this WPS server. </documentation>
        </annotation>
      </element>
    </sequence>
  </complexType>
</element>
<!-- ===== -->
<complexType name="ProcessSummaryType">
  <annotation>
    <documentation>
      The process summary consists of descriptive elements at the process
level,
      the process profiles and the service-specific properties.
      The process summary is not specific about process inputs and outputs.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="wps:DescriptionType">
      <attributeGroup ref="wps:processPropertiesAttributes"/>
    </extension>
  </complexContent>
</complexType>
</schema>

```

The schema for the supported deployment profiles is presented in the following listing:

supportedDeploymentProfiles.xsd

```

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:wps=
"http://www.opengis.net/wps/2.0" xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.opengis.net/wps/2.0" elementFormDefault="qualified"
version="2.0.0" xml:lang="en">
  <!-- ===== -->
  includes and imports
  ===== -->
  <!-- ===== -->
  elements and types
  ===== -->
  <!-- ===== -->
  <element name="SupportedDeploymentProfiles">
    <annotation>
      <documentation>List of default and other schemas profiles supported for
deployment and execution, i.e. ApacheOdeBPEL, GridProfile, ...</documentation>

```



```

</annotation>
<complexType>
  <sequence>
    <element name="Default">
      <annotation>
        <documentation>Identifies the default profile that will be
used unless the operation request specifies another supported profile.</documentation>
      </annotation>
      <complexType>
        <sequence>
          <element ref="wps:DeploymentSchema">
            <annotation>
              <documentation>Default deployment
schema</documentation>
            </annotation>
          </element>
        </sequence>
      </complexType>
    </element>
    <element name="Supported">
      <annotation>
        <documentation>Unordered list of references to all of the
profile schemas supported by this service. The default profile shall be included in
this list.</documentation>
      </annotation>
      <complexType>
        <sequence>
          <element ref="wps:DeploymentSchema" maxOccurs="unbounded
"/>
        </sequence>
      </complexType>
    </element>
  </sequence>
</complexType>
</element>
<!-- ===== -->
<element name="DeploymentSchema">
  <complexType>
    <attribute name="name" type="xs:normalizedString"/>
    <attribute name="href" type="anyURI"/>
  </complexType>
</element>
</schema>

```

The following listing shows the schema for the DeployProcess operation:

wpsDeployProcess.xsd

```

<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ows=
"http://www.opengis.net/ows/2.0" xmlns:wps="http://www.opengis.net/wps/2.0"
targetNamespace="http://www.opengis.net/wps/2.0" elementFormDefault="qualified"

```

```

version="2.0.0" xml:lang="en">
  <annotation>
    <documentation>
      WPS is an OGC Standard.
      Copyright (c) 2015 Open Geospatial Consortium.
      To obtain additional rights of use, visit
      http://www.opengeospatial.org/legal/.
    </documentation>
  </annotation>
  <!-- all-components document include (OGC 06-135r11 s#14) -->
  <include schemaLocation="wps.xsd"/>
  <!-- === IMPORTS === -->
  <import namespace="http://www.opengis.net/ows/2.0" schemaLocation=
"http://schemas.opengis.net/ows/2.0/owsAll.xsd"/>
  <import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation=
"http://www.w3.org/2001/xml.xsd"/>
  <!-- === INCLUDES === -->
  <!-- ===== -->
  <!-- DeployProcess request elements and types -->
  <!-- ===== -->
  <complexType name="ExecutionUnitType">
    <choice>
      <element ref="wps:Reference"/>
      <element name="Unit" type="anyType">
        <annotation>
          <documentation>Identifies the ExecutionUnit being encapsulated in
the DeployProcess request. If the provided document is a binary file, it is assumed
that it uses base64 encoding.</documentation>
        </annotation>
      </element>
    </choice>
  </complexType>
  <complexType name="DeploymentProfileType">
    <sequence/>
  </complexType>
  <element name="DeployProcess">
    <annotation>
      <documentation>WPS DescribeProcess operation request. </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base="wps:RequestBaseType">
          <sequence>
            <element name="ProcessDescription" type=
"wps:ProcessDescriptionChoiceType" minOccurs="0">
              <annotation>
                <documentation>Full description of process, including
all input and output parameters. This may be generated from the ExecutionUnit. The
document provides the informaiton intended to the WPS client.</documentation>
              </annotation>
            </element>

```

```

        <element name="ExecutionUnit" maxOccurs="unbounded">
            <annotation>
                <documentation>Concrete executable unit of the process
also called 'Application Package'. It may contains libraries, configuration files,
binaries, etc.</documentation>
            </annotation>
            <complexType>
                <complexContent>
                    <extension base="wps:ExecutionUnitType"/>
                </complexContent>
            </complexType>
        </element>
        <element name="DeploymentProfileName" type="normalizedString"
minOccurs="0">
            <annotation>
                <documentation>Name of the process profile deployment.
By default, the default WPS implementation process profile is
expected.</documentation>
            </annotation>
        </element>
    </sequence>
    <attribute ref="xml:lang" use="optional">
        <annotation>
            <documentation>RFC 4646 language code of the human-
readable text (e.g. "en-CA") in the process description.</documentation>
        </annotation>
    </attribute>
    <attribute name="immediateDeployment" type="boolean" default="
true">
        <annotation>
            <documentation>Deploy the ExecutionUnit on the backend
only on the first execution of the process.</documentation>
        </annotation>
    </attribute>
</extension>
</complexContent>
</complexType>
</element>
<!-- ===== -->
<!-- DeploymentProcess response elements and types -->
<!-- ===== -->
<element name="DeploymentResult">
    <annotation>
        <documentation>DeploymentProcess result.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="wps:DeploymentResultType"/>
        </complexContent>
    </complexType>
</element>

```

```

<complexType name="DeploymentResultType">
  <sequence>
    <element ref="ows:Identifier"/>
    <choice>
      <element name="ProcessSummary" type="wps:ProcessSummaryType"
minOccurs="0">
        <annotation>
          <documentation>Summary of the deployment process, if
relevant.</documentation>
        </annotation>
      </element>
    </choice>
  </sequence>
</complexType>
<complexType name="ProcessDescriptionChoiceType">
  <choice>
    <element ref="wps:Reference"/>
    <element ref="wps:ProcessOffering">
      <annotation>
        <documentation>Identifies process description as being
encapsulated in the DeployProcess request.</documentation>
      </annotation>
    </element>
  </choice>
</complexType>
</schema>

```

The following listing shows the schema for the UndeployProcess operation:

supportedDeploymentProfiles.xsd

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2016 rel. 2 sp1 (x64) (http://www.altova.com) by Joel Lim
(spacebel SA) -->
<!--
  OGC LICENSE HEADER

  The official OGC XML schemas can be accessed under http://schemas.opengis.net.
  Please consider the schema copyright notices:
http://www.opengeospatial.org/ogc/document
  and the OGC Copyright Notice and Disclaimers:
http://www.opengeospatial.org/ogc/legal
  If there are any questions left there is also a copyright FAQ:
http://www.opengeospatial.org/ogc/legalfaq
-->
<schema xmlns="http://www.w3.org/2001/XMLSchema" xmlns:ows=
"http://www.opengis.net/ows/2.0" xmlns:wps="http://www.opengis.net/wps/2.0"
targetNamespace="http://www.opengis.net/wps/2.0" elementFormDefault="qualified"
version="2.0.0" xml:lang="en">
  <annotation>
    <documentation>

```

WPS is an OGC Standard.

Copyright (c) 2015 Open Geospatial Consortium.

To obtain additional rights of use, visit

<http://www.opengeospatial.org/legal/>.

```
</documentation>
</annotation>
<!-- all-components document include (OGC 06-135r11 s#14) -->
<include schemaLocation="wps.xsd"/>
<!-- === IMPORTS === -->
<import namespace="http://www.opengis.net/ows/2.0" schemaLocation=
"http://schemas.opengis.net/ows/2.0/owsAll.xsd"/>
<import namespace="http://www.w3.org/XML/1998/namespace" schemaLocation=
"http://www.w3.org/2001/xml.xsd"/>
<!-- === INCLUDES === -->
<!-- ===== -->
<!-- UndeployProcess request elements and types -->
<!-- ===== -->
<element name="UndeployProcess">
  <annotation>
    <documentation>WPS UndeployProcess operation request.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="wps:RequestBaseType">
        <sequence>
          <element ref="ows:Identifier">
            <annotation>
              <documentation>Identifier of the process to be
undeployed.</documentation>
            </annotation>
          </element>
        </sequence>
        <attribute name="keepExecutionUnit" type="boolean" default="false
"/>
      </extension>
    </complexContent>
  </complexType>
</element>
<!-- ===== -->
<!-- UndeployProcess response elements and types -->
<!-- ===== -->
<element name="UndeploymentResult" type="wps:UndeploymentResultType">
  <annotation>
    <documentation>UndeployProcess result.</documentation>
  </annotation>
</element>
<complexType name="UndeploymentResultType">
  <sequence>
    <element ref="ows:Identifier">
      <annotation>
        <documentation>Identifier of the undeployed
```

```
process.</documentation>
      </annotation>
    </element>
  </sequence>
</complexType>
</schema>
```

Appendix C: Revision History

Table 20. Revision History

Date	Editor	Release	Primary clauses modified	Descriptions
May 30, 2018	B. Pross	.1	all	initial version
Jun 5, 2018	B. Pross	.1	5,6,7,8	Add section 5,6,7,8
Jun 10, 2018	B. Pross	.1	8	Add first version of BPMN DeploymentProfile
Jul 25, 2018	B. Pross	.1	all	Adjustments with respect to OWS Common 2.0 and WPS 2.0
Aug 28, 2018	B. Pross, C. Noel	.1	6	Incorporated comments from Spacebel
Sep 26, 2018	B. Pross, C. Noel	.1	Annex b	Incorporated comments from Spacebel
Sep 26, 2018	B. Pross, C. Noel	.1	Annex a	Added Abstract Test Suite
Nov 15, 2018	B. Pross	.1	2, 3, Annex c	Incorporated comments from Scott Serich
Dec 18, 2018	B. Pross	.1	1, 5	Incorporated comments from Carl Reed

Appendix D: Bibliography

1. Pross, B., Stasch, C.: OGC Testbed-13: Workflows Engineering Report. Open Geospatial Consortium, <http://docs.opengeospatial.org/per/17-029r1.html> (2018).