

= ZOO Project WPS compliancy testing =

In response to the presentation entitled "COMPLIANCE TESTING OF OPEN SOURCE SOFTWARE FOR WEB PROCESSING SERVICES" (as below)

http://2010.foss4g.org/presentations_show.php?id=3604

report named No. CIP-ICT-PSP-2009-3-250474 entitled "Web Processing Service Study" contributed by Bastian Schäffer from 52°North for the client/project : BRIdging Services, Information and Data for Europe (<http://www.briseide.eu/>).

Indeed, some strange behavior of the ZOO Kernel was pointed out in the presentation and report mentioned above. So ZOO Tribe decided to answer to this report by making his own report about ZOO Project compliancy testing using the same way as in the report.

First note about the report we are currently writing, we weren't able to use XMLSpy 2007 Professional version as no more Evaluation version is available, so we used the XMLSpy 2011 Professional one as we can use it for an evaluation period of 30 days.

All the following requests give documents which are valid (using XMLSpy 2011 Professional) against the used schemas.

== !GetCapabilities request ==

In the section 5.5 of the report an error was pointed out about some unexpected metadata : namely Test attribute to the Metadata node getting the "Demo" value. This was due to wrong ZCFG files of the first Services we made in the beginning of the 2009 year .

The right URL to run this test is wrong, should be tested with the right one below:

http://zoo-project.org/cgi-bin-new/zoo_loader.cgi?REQUEST=GetCapabilities&SERVICE=WPS&version=1.0.0

The annex II linked to this section contains strangely a !ProcessDescription document when it shall be a Capabilities Document instead. Maybe we do not have the final

version of the report (at least ZOO-Team hope so).

So for the !GetCapabilities request, the Capabilities document output by the ZOO Kernel is "Valid".

== !DescribeProcess request ==

In the section 5.11 of the report we can read that the ZOO Kernel cannot handle the !DescribeProcess request and output a valid !ProcessDescriptions Document. Indeed the first versions of ZOO Kernel was wrong in that, but hopefully the last one, included in OSGeoLiveDVD 4.0 returns already the valid !ProcessDescriptions document.

http://zoo-project.org/cgi-bin-new/zoo_loader.cgi?REQUEST=DescribeProcess&SERVICE=WPS&version=1.0.0&Identifier=Buffer

Using the URL above we can confirm that ZOO Kernel is "Valid" for the !DescribeProcess request.

Note that this request is "Valid" also :

http://zoo-project.org/cgi-bin-new/zoo_loader.cgi?REQUEST=DescribeProcess&SERVICE=WPS&version=1.0.0&Identifier=Buffer,Boundary,Centroid,ConvexHull,Simplify,Union,Intersection,Difference,SymDifference

== Execute request ==

In the 5.13 section of the report you can get the requests for Execute as XML POST requests, as it is the mandatory method.

To use the XML POST requests provided in the report or easier by using copy/paste from this page, please follow this [http://zoo-project.org/test_services.html link].

=== section 5.17 (Synchronous call using asReference) ===

```
{{  
#!xml
```

```

<wps:Execute service="WPS" version="1.0.0"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
<ows:Identifier>Buffer</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>InputPolygon</ows:Identifier>
<ows:Title>Playground area</ows:Title>
<wps:Reference
xlink:href="http://dreal-official.geolabs.fr/mapjax/webservices/wfs/dreal_lr_general/?VE
RSION=1.1.0&version=1.0.0&request=GetFeature&typename=Znieff1&
&maxfeatures=1"/></wps:Input>
<wps:Input>
<ows:Identifier>BufferDistance</ows:Identifier>
<ows:Title>Distance which people will walk to get to a playground.</ows:Title>
<wps>Data>
<wps:LiteralData>10</wps:LiteralData>
</wps>Data>
</wps:Input>
</wps>DataInputs>
<wps:ResponseForm>
<wps:ResponseDocument storeExecuteResponse="false">
<wps:Output asReference="true">
<ows:Identifier>Result</ows:Identifier>
</wps:Output>
</wps:ResponseDocument>
</wps:ResponseForm>
</wps:Execute>
}}

```

=== section 5.18.1.4 (Synchronous call with data included) ===

```

{{{
#!xml
<wps:Execute service="WPS" version="1.0.0"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
<ows:Identifier>Buffer</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>InputPolygon</ows:Identifier>
<ows:Title>Playground area</ows:Title>
<wps:Reference
xlink:href="http://dreal-official.geolabs.fr/mapjax/webservices/wfs/dreal_lr_general/?VE
RSION=1.1.0&version=1.0.0&request=GetFeature&typename=Znieff1&
amp;maxfeatures=1"/></wps:Input>
<wps:Input>
<ows:Identifier>BufferDistance</ows:Identifier>
<ows:Title>Distance which people will walk to get to a playground.</ows:Title>
<wps>Data>
<wps:LiteralData>10</wps:LiteralData>
</wps>Data>
</wps:Input>
</wps>DataInputs>
<wps:ResponseForm>
<wps:ResponseDocument storeExecuteResponse="false">
<wps:Output asReference="false">
<ows:Identifier>Result</ows:Identifier>
</wps:Output>
</wps:ResponseDocument>
</wps:ResponseForm>
</wps:Execute>
}}}
```

=== section 5.23 (Asynchronous call with data asReference) ===

```
{{
#!xml
<wps:Execute service="WPS" version="1.0.0"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
<ows:Identifier>Buffer</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>InputPolygon</ows:Identifier>
<ows:Title>Playground area</ows:Title>
<wps:Reference
xlink:href="http://dreal-official.geolabs.fr/mapjax/webservices/wfs/dreal_lr_general/?VE
RSION=1.1.0&version=1.0.0&request=GetFeature&typename=Znieff1&
amp;maxfeatures=1"/></wps:Input>
<wps:Input>
<ows:Identifier>BufferDistance</ows:Identifier>
<ows:Title>Distance which people will walk to get to a playground.</ows:Title>
<wps>Data>
<wps:LiteralData>10</wps:LiteralData>
</wps>Data>
</wps:Input>
</wps>DataInputs>
<wps:ResponseForm>
<wps:ResponseDocument storeExecuteResponse="true" status="true">
<wps:Output asReference="true">
<ows:Identifier>Result</ows:Identifier>
</wps:Output>
</wps:ResponseDocument>
</wps:ResponseForm>
</wps:Execute>
```

}}

Here both statusLocation file and response are "Valid".

=== Section Not Found (Asynchronous call with full data included) ===

Even if it is not mentioned in the original report, you can also use the storeExecuteReponse="true" for !ResponseDocument with the asReference attribute for the Output settled to true as in the following request :

{{

#!/xml

```
<wps:Execute service="WPS" version="1.0.0"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
<ows:Identifier>Buffer</ows:Identifier>
<wps:DataInputs>
<wps:Input>
<ows:Identifier>InputPolygon</ows:Identifier>
<ows:Title>Playground area</ows:Title>
<wps:Reference
xlink:href="http://dreal-official.geolabs.fr/mapjax/webservices/wfs/dreal_lr_general/?VE
RSION=1.1.0&version=1.0.0&request=GetFeature&typename=Znieff1&
amp;maxfeatures=1"/></wps:Input>
<wps:Input>
<ows:Identifier>BufferDistance</ows:Identifier>
<ows:Title>Distance which people will walk to get to a playground.</ows:Title>
<wps>Data>
<wps:LiteralData>10</wps:LiteralData>
</wps>Data>
</wps:Input>
</wps>DataInputs>
```

```

<wps:ResponseForm>
<wps:ResponseDocument storeExecuteResponse="true" status="true">
<wps:Output asReference="false">
<ows:Identifier>Result</ows:Identifier>
</wps:Output>
</wps:ResponseDocument>
</wps:ResponseForm>
</wps:Execute>
}}}

```

The resulting document and the statusLocation document are both "Valid".

=== section 5.28 (!RawDataOutput test) ===

```

{{{
#!xml
<wps:Execute service="WPS" version="1.0.0"
xmlns:wps="http://www.opengis.net/wps/1.0.0"
xmlns:ows="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wps/1.0.0
http://schemas.opengis.net/wps/1.0.0/wpsExecute_request.xsd">
<ows:Identifier>Buffer</ows:Identifier>
<wps>DataInputs>
<wps:Input>
<ows:Identifier>InputPolygon</ows:Identifier>
<ows:Title>Playground area</ows:Title>
<wps:Reference
xlink:href="http://dreal-official.geolabs.fr/mapjax/webservices/wfs/dreal_lr_general/?VE
RSION=1.1.0&version=1.0.0&request=GetFeature&typename=Znieff1&
amp;maxfeatures=1"/></wps:Input>
<wps:Input>
<ows:Identifier>BufferDistance</ows:Identifier>
<ows:Title>Distance which people will walk to get to a playground.</ows:Title>
<wps>Data>

```

```
<wps:LiteralData>10</wps:LiteralData>
</wps:Data>
</wps:Input>
</wps:DataInputs>
<wps:ResponseForm>
<wps:RawDataOutput>
<ows:Identifier>Result</ows:Identifier>
</wps:RawDataOutput>
</wps:ResponseForm>
</wps:Execute>
}}
```

The result is valid in the sense that it is a JSON string representing the resulting polygon.