INSPIRE
Infrastructure for Spatial Information in Europe

Draft Technical Guidance for INSPIRE Coordinate Transformation Services
Drafting Team "Network Services"

Title	Draft Technical Guidance for INSPIRE Coordinate Transformation Services
Creator	Network Services Drafting Team
Date	2010-03-15
Subject	Technical Guidance for INSPIRE Coordinate Transformation Services
Status	Draft
Publisher	Network Services Drafting Team
Type	Text
Description	This document defines technical guidance for INSPIRE Coordinate Transformation Services according to the Transformation Services IR
Format	MS Word (doc)
Source	Network Services Drafting Team
Rights	INSPIRE DT members
Identifier	Draft Technical Guidance Coordinate Transformation Services v2.1.doc
Language	EN
Relation	Not applicable
Coverage	Project duration

Table of contents

INSPIRE Coordinate Transformation Service as a WCTS-based AP of the WPS

1 Introduction

This document defines the technical service interface for the INSPIRE Coordinate Transformation Service. The definition is based on the conceptual model established in the INSPIRE Implementing Rule (IR) for Transformation Services. There isn't any official, internationally agreed service interface specification available for coordinate transformations. The approach adopted in this document is to incorporate the only mandatory operation `Transform` defined in of the OGC Discussion Paper 'Web Coordinate Transformation Service' (WCTS) as an Application Profile of the OGC Implementation Specification 'Web Processing Service' (WPS). The `IsTransformable` operation of the WCTS is implemented as an optional parameter of the main transformation operation. All other optional operations of the WCTS interface are omitted.

The OGC's WCTS specification is available as an OGC Discussion Paper (version 0.4.0). There seems to be no activity in the OGC to advance the WCTS specification to official level. The development of the few implementations of the specification has mostly been discontinued.

The WPS interface standard is an official OGC Implementation Specification (version 1.0.0). WPS defines a generic Web-based geospatial processing service interface. Specific processes can be specified as Application Profiles (AP) of the WPS. As the trend in the OGC Web service interface development seems to be towards the use of a generic processing service access interface, it has been decided to define the INSPIRE Coordinate Transformation Service by specifying the WCTS's `Transform` operation as an AP of the WPS specification.

In the following discussion 'TransformCoordinates' is used as the value of the WPS process identifier; however this value is internal to the service and can be arbitrary. A conforming process is identified by the URN of the corresponding WPS AP.

This document does not detail the recommended SOAP framework for implementing the Coordinate Transformation Service, see

http://inspire.jrc.ec.europa.eu/reports/ImplementingRules/network/INSPIRE_NETWORK_SERVICES_SOAP_Framework.pdf

2 Normative references

The following referenced documents are indispensable for the application 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.

IETF RFC 2616 Hypertext Transfer Protocol -- HTTP/1.1

IETF RFC 2818 HTTP Over TLS

OGC 07-055r1 Web Coordinate Transformation Service 0.4.0

OGC 05-007r7 Web Processing Service 1.0.0

ISO 19136:2007 Geographic information -- Geography Markup Language (GML)

OGC 05/095r1 GML 3.1.1 common CRSs profile 1.0.0

ISO 19119*:2005, Geographic information – Services*

ISO 19119:2005 PDAM 1, *Geographic information – Services*

OGC 06-121r3 **OGC OWS** OpenGIS Web Service Common Implementation Specification 1.1.0

INSPIRE, **INS ARC**, Network Services Architecture Version 3.0

3   Abbreviations

AP	Application Profile
CRS	Coordinate Reference System
EPSG	European Petroleum Survey Group, currently OGP Surveying and Positioning Committee
GML	Geography Markup Language
INSPIRE	Infrastructure for Spatial Information in Europe
IR	Implementing Rule
JRC	Joint Research Centre
MIME	Multipurpose Internet Mail Extensions
OGC	Open Geospatial Consortium
OGP	International Association of Oil & Gas Producers
URL	Uniform Resource Locator
URN	Uniform Resource Name
WCTS	Web Coordinate Transformation Service
WFS	Web Feature Service
WPS	Web Processing Service
XML	Extensible Markup Language

4 WCTS's Transform -operation as an AP of WPS

WPS specification defines three operations: `GetCapabilities`, `DescribeProcess` and `Execute`. The idea in the service interface is that the processes supported by a given WPS implementation are listed in the `GetCapabilities` response message. A client application can select a process identifier from that list and request more detailed metadata about this process by the `DescribeProcess` operation. The response message of the `DescribeProcess` operation contains textual description about what the process does and lists all the input parameters, together with details of their data type and allowed values. Finally, based on received information, the client can construct an appropriate process request and send it to the service via the `Execute` operation.

The operations defined at the public service interface of the WCTS can be treated as processes of the WPS. So, for instance, the `Transform` operation of the WCTS shall become a process `TransformCoordinates` of the WPS. The two main principles of this service interface mapping are:

  a) The essential service metadata content as defined in the `GetCapabilities` response of the WCTS will be incorporated into the `DescribeProcess` response message of the WPS
  b) The input parameters defined for the WCTS `Transform` operation are applied as input parameters of the WPS `TransformCoordinates` process in a straightforward manner

Only the two mandatory operations of the WCTS, `GetCapabilities` and `Transform`, are considered.

4.1 Service Metadata in WCTS

The service metadata present in the response message of the WCTS `GetCapabilities` query is divided into sections: `ServiceIdentification`, `ServiceProvider`, `OperationsMetadata` and `Content`. `ServiceIdentification` and `ServiceProvider` contain generic information of the service and the service provider, as defined in the OGC Web Service Common Specification. These metadata elements are common to all OGC services. In the `OperationsMetadata` section all the public operations of the service are listed, together with the URL from which the operation can be found. In the case of a WCTS server, this includes the `GetCapabilities` and `Transform` operations, and all the optional WCTS operations that the server implements.

In the `Contents` section the following general metadata elements are present: `Transformation`, `Method`, `SourceCRS`, `TargetCSR`, `FeatureAbilities/GeometryType`, `FeatureAbilities/FeatureFormat`. These give, respectively, the well-known coordinate transformations, well-known coordinate operation methods (part of user defined coordinate operations), source and target CRSs, and the GML geometry types and the MIME-type of the input data content supported by the WCTS server.

4.2 Service Metadata in WPS

In the WPS `CetCapabilities` response message the service can list all the available process names under the element: `ProcessOfferings/Process/Identifier`. Also a title, an abstract and a single textual metadata record can be given. In the `Profile` element an identification of the used WPS AP can be given.

When finding an interesting process the client application can send further request (`DescribeProcess`) for additional process-specific metadata. In the response message of this request the following process metadata elements are present: `ProcessDescriptions/ProcessDescription/DataInputs/Input` and

`ProcessDescriptions/ProcessDescription/ProcessOutputs/Output`. Both of them have nearly identical content and contain, in addition to the `Identifier`, `Title`, `Abstract` and `Metadata` elements, a description of the allowed input and output parameter data types.

The main data type categories are ComplexData and LiteralData (`ComplexOutput`, `LiteralOutput`). `ComplexData` (`ComplexOutput`) refers to XML-encoded structural input/output parameters. For these an indication about the allowed/available `MimeType`, `Encoding` and `Schema` are given. `LiteralData` (`LiteralOutput`) refers to parameters expressed as literal data types. For these, allowed values can be given as lists or ranges.

4.3 WCTS service metadata in the WPS DescribeProcess operation

Metadata elements of the WCTS `GetCapabilities` operation can be incorporated into the WPS `DescribeProcess` (identifier: `TransformCoordinates`) response message in the following way.

`Transformation`, `SourceCRS` and `TargetCRS` can be introduced as literal input parameters for the `TransformCoordinates` process (`Transformation` and `SourceCRS`/`TargetCRS` are mutually exclusive; however, in the case `Transformation` represents a conversion, `TargetCRS` must be provided). They can be expressed as URN strings.Allowed values can be listed in the `LiteralData/AllowedValues` data structure of the `DescribeProcess` response message, corresponding to the `Transformation`, `SourceCRS` and `TargetCRS` metadata elements of the `WCTS GetCapabilities` operation response.

The `FeatureAbilities` metadata element will be covered by data type information related to the input parameter `InputData`. The value of this parameter is the GML-encoded dataset to be transformed. In the `DescribeProcess` response message the allowed schemas for the input dataset can be indicated (`ComplexData/Supported/Format/Schema`). The supported GML geometry types are those defined in this schema. The supported MIME-type can be given in the element `ComplexData/Supported/Format/MimeType`. The default values used by the service are indicated in `ComplexData/Default/Format/…`). The mapping from the WCTS metadata elements to the WPS `DescribeProcess` response elements is shown in Table 1.

The URN identifier of the used WPS AP must be present in the `DescribeProcess` response message.

WCTS metadata (GetCapabilities)	WPS metadata (DescribeProcess)
`Transformation`	`DataInputs/Input(Identifier=Transformation)/ LiteralData/AllowedValues/Value`
`Method`	Not applicable, because user-defined transformations are not supported
`SourceCRS`	`DataInputs/Input(Identifier=SourceCRS)/ LiteralData/AllowedValues/Value`
`TargetCRS`	`DataInputs/Input(Identifier=TargetCRS)/ LiteralData/AllowedValues/Value`
`FeatureAbilities/ GeometryType`	`DataInputs/Input(Identifier=InputData)/ ComplexData/Supported/Format/Schema`
`FeatureAbilities/ FeatureFormat`	`DataInputs/Input(Identifier=InputData)/ ComplexData/Supported/Format/MimeType`

Table 1. The metadata elements of the WCTS mapped to the elements of the WPS `DescribeProcess` response

4.4 Process execution

In addition to the service metadata query, `GetCapabilities`, the only mandatory operation defined in the WCTS is the `Transform` operation. By this operation a calling application can indicate the source and target CRSs, or a well-known coordinate transformation, and provide the input dataset to be transformed. In the WPS interface all processed are called through the generic `Execute` operation. When defining the WCTS `Transform` operation as a process `TransformCoordinates` in the WPS interface, the corresponding input parameters can be incorporated in a straightforward manner.

In the `Execute/Identifier` element an indication of the called process (e.g. 'TransformCoordinates') can be given. The WCTS input parameters `SourceCRS`, `TargetCRS` and `InputData` become the three Input elements of the WPS `Execute(TransformCoordinates)` call. When input dataset is not inline, a reference can be given to a Web-accessible resource. In the WCTS `Transform` operation this is done by the element `Transform/InputData/Reference` and in WPS Execute operation by the element `Execute/DataInputs/Input/Reference`.

The call can also request certain output format to be applied. In the WCTS this is given as a MIME type in the element `OutputFormat`. In the WPS `Execute` request the same can be done by the element `ResponseForm/ResponseDocument/Output/@mimeType`, with the additional capability to also indicate the relevant schema.

WCTS Transform	WPS Execute(TransformCoordinates)
`Transformation`	`DataInputs/Input(Identifier=Transformation)/ Data/LiteralData`
`SourceCRS`	`DataInputs/Input(Identifier=SourceCRS)/ Data/LiteralData`
`TargetCRS`	`DataInputs/Input(Identifier=TargetCRS)/ Data/LiteralData`
`InputData`	`DataInputs/Input(Identifier=InputData)/ Data/ComplexData`
`InputData/ Reference`	`DataInputs/Input(Identifier=InputData)/ Reference`
`OutputFormat`	`ResponseForm/ResponseDocument/Output/@mimeType`

Table 2. The input parameters of the WCTS mapped to the parameters of the WPS operation Execute(TransformCoordinates)

5 Mapping to the INSPIRE Transformation Service IR

The presented Coordinate Transformation Service interface conforms to the abstract service definition established in the INSPIRE Implementing Rules (IR) for Transformation Services. In the Tables 3 and 4 the mapping is presented from the abstract operations introduced in the IR to the service interface elements described in this document.

Abstract operation	WPS operation
Get Transformation Service Metadata	**GetCapabilities** and **DescribeProcess(TransformCoordinates)**
Transform	**Execute(TransformCoordinates)**
Link Transformation Service	To be implemented by uploading the required metadata to an INSPIRE-compliant Discovery Service using the PublishMetadata operation

Table 3. Mapping from the operations defined in the Transformation Service IR to the concrete operations defined in this Technical Guidance

5.1 Get Transformation Service Metadata operation

The natural language requested for the response of the Get Transformation Service Metadata operation shall be given using the language parameter of the WPS GetCapabilities request as described in the Chapter 8.2 of the WPS 1.0.0 specification.

The Transformation Service Metadata response parameter of the Get Transformation Service Metadata operation shall be given using the Service Identification and Service Provider parameters of the WPS GetCapabilities response as described in the Chapter 8.3 of the WPS 1.0.0 specification.

The Operations Metadata response parameter of the Get Transformation Service Metadata operation shall be given using the WPS DescribeProcess(TransformCoordinates) response data structure as described in the Chapter 9 of the WPS 1.0.0 specification. The supported transformation category (coordinate transformation) is indicated by the Profile parameter of the ProcessDescription data structure. The accepted encoding of the Input Spatial Data Set are indicated by the corresponding Format parameter of the Input data structure. The parameters related to the accepted Data Model languages and Model Mapping languages are not relevant, because in the case of the Coordinate Transformation Service these parameters are restricted to literal identifiers as described below.

The Languages response parameter of the Get Transformation Service Metadata operation shall be given using the lang and Languages parameter of the WPS GetCapabilities response as described in the Chapter 8.3 of the WPS 1.0.0 specification. The lang parameter is used to indicate the natural

language used in the Get Transformation Service Metadata response and the `Language` parameter is used to list the languages supported by the service.

5.2 Transform operation

Abstract parameter	WPS element
Input Spatial Data Set	`Execute/DataInputs/Input/ComplexData,` or `Execute/DataInputs/Input/Reference,` where `Input/Identifier = 'InputData'`
Source Model	`Execute/DataInputs/Input/Data/LiteralData,` where `Input/Identifier = 'SourceCRS'`
Target Model	`Execute/DataInputs/Input/Data/LiteralData,` where `Input/Identifier = 'TargetCRS'`
Model Mapping	`Execute/DataInputs/Input/Data/LiteralData,` where `Input/Identifier = 'Transformation'`

Table 4. Mapping from the parameters of the TRANSFORM –operation defined in the Transformation Service IR to the elements defined in this Technical Guidance.

The value of the Input Spatial Data Set –parameter shall be given to the process either as an inline XML-structure (`ComplexData`) or as a reference (URL) to a Web-accessible resource (`Reference`, typically a WFS `GetFeature` –query). The service is required to support all INSPIRE-defined GML APs as an input and output data schema. The input and output shall always reference to the same schema.

The GML input might also contain coverage data, encoded in the form of a GML feature. In this case the desired interpolation method can be indicated by an optional input parameter `InterpolationMethod (LiteralData)`.

The value of the Source Model and Target Model –parameters shall be given to the process as a well-known CRS identifier (`LiteralData`, URN) The CRS identifier shall be given in the following form, as specified in the OGC Best Practice document 'Definition identifier URNs in OGC namespace':

urn:ogc:def:crs:[AUTHORITY]:[DB-VERSION]:[CODE]

The database version can be omitted, in which case the latest available CRS definition is assumed. The input parameter 'SourceCRS' can be omitted, if the input data set contains indication of the CRS used ('srsName' parameter in GML-encoded input). The input data set is supposed to contain geometries only in one CRS. If the request identifies the used source CRS both by the 'SourceCRS' parameter and inside the input data set, the given values must be consistent.

The value of the Model Mapping –parameter shall be given to the process as a well-known transformation identifier (`LiteralData`, URN). The transformation identifier shall be given in the same form as the identifier for CRS, with 'crs' replaced by 'coordinateOperation'.

An additional boolean parameter **TestTransformation** is introduced for the **Execute(TransformCoordinates)** operation. This is an optional parameter and facilitates testing whether a service can perform the indicated request. If the value of the parameter is '**false**' or it is missing from the request the service is supposed to perform an ordinary coordinate transformation. If the value is '**true**' the service is expected to only test if it is possible to carry out the transformation. The result of the testing is returned in an exception message with the **ExceptionCode** having the value of '**Transformable**' in the case of a successful transformation and value '**NotTransformable**' in the case transformation is not possible. When returning the value '**NotTransformable**' the service can indicate the reason for the failure in the **ExceptionText** element.

6 WPS Application Profile for INSPIRE Coordinate Transformation Service

According to the WPS specification, the Application Profile (AP) of the standard is defined by

   a) an OGC URN that uniquely identifies the process
   b) a reference response message to the **DescribeProcess** request for that process.

A human-readable documentation that describes the process may also be provided.

The OGC URN of the coordinate transformation process of the WPS-based INSPIRE Coordinate Transformation Service is:

urn:ogc:wps:1.0.0:INSPIRE:TransformCoordinates:1.0

(or urn:x-inspire:specification:TransformationService:CoordinateTransformation:1.0)

A reference **DescribeProcess** response is given in Appendix A.

An exemplary **Execute** query can be found in Appendix B.

The exception codes introduced in this AP are listed in Appendix C.

7 Quality of Service Requirements

7.1 Performance

Output performance for a Coordinate Transformation Service carrying out a simple map projection on an input data set consisting of GML-encoded feature data with only geometric properties included shall be 1 MB/s with an initial response time of 0.5 s/MB (of input data).

7.2 Capacity

Transformation Service is required to support 5 requests per second.

7.3 Availability

The probability of the Transformation Service to be available shall be 99% of the time.

Appendix A

In the following a reference response message to the **DescribeProcess** call for the **TransformCoordinates** process is given. The element values to be followed in letter are indicated in blue colour.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--This example describes a coordinate transformation process that accepts
coordinates encoded as GML Simple Features profile compliant features.  The
results can be returned directly as output, or stored by the service as a
web-accessible resource.  Ongoing processing status reports are not
available. -->

<wps:ProcessDescriptions
    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
    ..\wpsDescribeProcess_response.xsd" service="WPS" version="1.0.0"
    xml:lang="en">

   <ProcessDescription wps:processVersion="2" storeSupported="true"
statusSupported="false">
      <ows:Identifier>TransformCoordinates</ows:Identifier>
      <ows:Title>Transform coordinates.</ows:Title>
      <ows:Abstract>Transforms coordinates from a given source CRS to the
indicated target CRS. Accepts input as GML Simple Features encoded features
and provides the same encoding as output.</ows:Abstract>
      <ows:Metadata xlink:title="coordinate transformation"/>
      <ows:Metadata xlink:title="coordinate conversion"/>
      <ows:Metadata xlink:title="GML"/>

   <wps:Profile>urn:ogc:wps:1.0.0:INSPIRE:TransformCoordinates:1.0</wps:Pro
file>

     <DataInputs>
       <Input minOccurs="0" maxOccurs="1">
         <ows:Identifier>SourceCRS</ows:Identifier>
         <ows:Title>Source Coordinate Reference System</ows:Title>
         <ows:Abstract>Indication of the CRS the input dataset is presented
in.</ows:Abstract>
         <LiteralData>
          <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#anyURI">anyURI</ows:DataType>
          <ows:AllowedValues>
          <!-- Given here only as an example, implementing services define
the values according to their capabilities -->
          <ows:Value>urn:ogc:def:crs:EPSG:6.12:4258</ows:Value>
          <ows:Value>urn:ogc:def:crs:EPSG:6.12:3386</ows:Value>
          <ows:Value>urn:ogc:def:crs:EPSG:6.12:2391</ows:Value>
          <ows:Value>urn:ogc:def:crs:EPSG:6.12:2392</ows:Value>
          <ows:Value>urn:ogc:def:crs:EPSG:6.12:2393</ows:Value>
          <ows:Value>urn:ogc:def:crs:EPSG:6.12:2394</ows:Value>
          <ows:Value>urn:ogc:def:crs:EPSG:6.12:3387</ows:Value>
         </ows:AllowedValues>
         <DefaultValue>urn:ogc:def:crs:EPSG:6.12:2393</DefaultValue>
        </LiteralData>
      </Input>
      <Input minOccurs="0" maxOccurs="1">
        <ows:Identifier>TargetCRS</ows:Identifier>
        <ows:Title>Target Coordinate Reference System</ows:Title>
        <ows:Abstract>Indication of the CRS the output dataset is requested
in.</ows:Abstract>
        <LiteralData>
```

```
        <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#anyURI">anyURI</ows:DataType>
        <ows:AllowedValues>
        <!-- Given here only as an example, implementing services define
the values according to their capabilities -->
        <ows:Value>urn:ogc:def:crs:EPSG:6.12:3067</ows:Value>
        </ows:AllowedValues>
        <DefaultValue>urn:ogc:def:crs:EPSG:6.12:3067</DefaultValue>
        </LiteralData>
      </Input>
      <Input minOccurs="0" maxOccurs="1">
        <ows:Identifier>TestTransformation</ows:Identifier>
        <ows:Title>Operation mode</ows:Title>
        <ows:Abstract> Indication of the mode of the operation: perform
real transformation if TestTransformation is 'false,' only test
transformation if 'true'.</ows:Abstract>
        <LiteralData>
        <ows:DataType ows:reference="http://www.w3.org/TR/xmlschema-
2/#boolean">boolean</ows:DataType>
        <ows:AllowedValues>
          <ows:Value>true</ows:Value>
          <ows:Value>false</ows:Value>
        </ows:AllowedValues>
        <DefaultValue>false</DefaultValue>
        </LiteralData>
      </Input>
      <Input minOccurs="1" maxOccurs="1">
        <ows:Identifier>InputData</ows:Identifier>
        <ows:Title>GML SF-encoded input dataset</ows:Title>
        <ows:Abstract>Input dataset as GML Simple Features compliant
features.</ows:Abstract>
        <ComplexData maximumMegabytes="5">
        <!-- Given here only as an example, implementing services define
the values according to their capabilities. The services are required to
support all INSPIRE theme-specific GML APs. -->
        <Default>
          <Format>
          <MimeType>application/gml+xml; version=3.1</MimeType>
          <Encoding>UTF-8</Encoding>
          <Schema>http://geoinfo.fgi.fi/schema/Cpoints.xsd</Schema>
          </Format>
        </Default>
        <Supported>
          <Format>
          <MimeType>application/gml+xml; version=3.1</MimeType>
          <Encoding>UTF-8</Encoding>
          <Schema>http://geoinfo.fgi.fi/schema/Cpoints.xsd</Schema>
          </Format>
          <Format>
          <MimeType>application/gml+xml; version=3.1</MimeType>
          <Encoding>UTF-8</Encoding>
<Schema>http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/gm
lsf.xsd</Schema>
          </Format>
        </Supported>
        </ComplexData>
      </Input>
    </DataInputs>

    <ProcessOutputs>
      <Output>
        <ows:Identifier>TransformedData</ows:Identifier>
        <ows:Title>Transformed dataset</ows:Title>
        <ows:Abstract>GML stream containing the resulting transformed
dataset.</ows:Abstract>
        <ComplexOutput>
        <!-- Given here only as an example, implementing services define
the values according to their capabilities. The services are required to
support all INSPIRE theme-specific GML APs. -->
```

```
          <Default>
            <Format>
              <MimeType>application/gml+xml; version=3.1</MimeType>
              <Encoding>UTF-8</Encoding>
              <Schema>http://geoinfo.fgi.fi/schema/Cpoints.xsd</Schema>
            </Format>
          </Default>
          <Supported>
            <Format>
              <MimeType>application/gml+xml; version=3.1</MimeType>
              <Encoding>UTF-8</Encoding>
              <Schema>http://geoinfo.fgi.fi/schema/Cpoints.xsd</Schema>
            </Format>
            <Format>
              <MimeType>application/gml+xml; version=3.1</MimeType>
              <Encoding>UTF-8</Encoding>
<Schema>http://schemas.opengis.net/gml/3.1.1/profiles/gmlsfProfile/1.0.0/gm
lsf.xsd</Schema>
            </Format>
          </Supported>
        </ComplexOutput>
      </Output>
    </ProcessOutputs>
  </ProcessDescription>
</wps:ProcessDescriptions>
```

Appendix B

In the following an exemplary **Execute** request for process **TransformCoordinates** is presented.

```xml
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!-- Execute operation request for coordinate transformation.-->

   <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
     ..\wpsExecute_request.xsd">

   <ows:Identifier>TransformCoordinates</ows:Identifier>
   <wps:DataInputs>
     <wps:Input>
       <ows:Identifier>SourceCRS</ows:Identifier>
       <wps:Data>
         <wps:LiteralData>urn:ogc:def:crs:EPSG:6.12:2392</wps:LiteralData>
       </wps:Data>
     </wps:Input>
     <wps:Input>
       <ows:Identifier>TargetCRS</ows:Identifier>
       <wps:Data>
         <wps:LiteralData>urn:ogc:def:crs:EPSG:6.12:3067</wps:LiteralData>
       </wps:Data>
     </wps:Input>
     <wps:Input>
       <ows:Identifier>InputData</ows:Identifier>
       <wps:Data>
         <wps:ComplexData>
          <cp:Cpoints
            xmlns:cp="http://www.fgi.fi/Cpoints"
            xmlns:gml="http://www.opengis.net/gml"
            xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:schemaLocation="http://www.fgi.fi/Cpoints
http://geoinfo.fgi.fi/schema/Cpoints.xsd">
           <gml:boundedBy>
             <gml:Envelope srsName="urn:ogc:def:crs:EPSG:6.12:2392">
               <gml:lowerCorner/>
               <gml:upperCorner/>
             </gml:Envelope>
           </gml:boundedBy>
           <cp:featureMembers>
             <cp:Cpoint>
               <cp:location>
                 <gml:Point>
                   <gml:pos>6798345.234 2546976.432</gml:pos>
                 </gml:Point>
               </cp:location>
             </cp:Cpoint>
             <cp:Cpoint>
               <cp:location>
                 <gml:Point>
                   <gml:pos>6798465.085 2546832.973</gml:pos>
                 </gml:Point>
               </cp:location>
             </cp:Cpoint>
           </cp:featureMembers>
```

```
          </cp:Cpoints>
        </wps:ComplexData>
      </wps:Data>
    </wps:Input>
  </wps:DataInputs>

  <wps:ResponseForm>
    <wps:ResponseDocument storeExecuteResponse="true" lineage="false"
status="true">
      <wps:Output asReference="true">
        <ows:Identifier>TransformedData</ows:Identifier>
      </wps:Output>
    </wps:ResponseDocument>
  </wps:ResponseForm>

</wps:Execute>
```

Appendix C

The Exception codes applicable in the INSPIRE Coordinate Transformation Service, in addition to the generic Exception codes specified in the WPS base document:

Exception Code	Description	ExceptionText
NoInputData	No input data available from a specified source for the input	Value of the xlink:href attribute in InputData
InvalidArea	One or more points are outside the domainOfValidity of the coordinate operation	Value of the xlink:href attribute in InputData
TransformException	Returned in case of an computation error occuring during an coordinate operation	Omit
OperationNotSupported	Request is for an operation that is not supported by this service instance	Name of operation not supported
UnsupportedCombination	Operation request contains an output CRS, which can not be used within the output format	Name of the Format/CRS
Transformable	The coordinate transformation with the defined parameters is possible. Not an actual exception!	Omit
NotTransformable	The coordinate transformation can not be performed	A specific reason for the exception, like the geometries were outside the target CRS's valid area
MutualExclusionException	The multiplicities of the input parameters are mutually in contradiction	Identifiers of the parameters