Earth Observation Data Management System (EODMS) for Department of Natural Resources Canada

User's Guide

Revision 1.1

8.0 WES OGC INTERFACE

The WES OGC Interface is made up of a number of OGC standard services including CSW (2.0.2) and WFS (2.0.0, 1.1.0) services for searching and WCS (2.0.1, 2.0.0, 1.1.2, 1.1.1) and WPS services for ordering.

These service interfaces provide an alternate way for both users and developers to search and order from the EODMS catalog without having to login through the EODMS web application. The sections below outline how to use each of these service interfaces.

8.1 Authentication

In order to use the WES OGC WCS, CSW, WFS or the WPS service interfaces, users must specify their credentials in each request. The only endpoints that do not require credentials are the CSW endpoints: EODMS CSW Catalog and National Air Photo Library. See Section 8.2.4 for more details about each available EODMS CSW endpoint.

For WCS, WFS and the WPS this must be done via basic authentication in the HTTP request header. For CSW, this can be done via basic authentication or with SOAP through the WSS (Web Service Security) extension.

To use basic authentication, users must specify **Authorization** as the header name and Basic username:password as the value where username:password is Base64 encoded.

Example:

Authorization Basic billc:billc_pw1

Once the username:password is converted to Base64, this becomes:

Authorization Basic YmlsbGM6YmlsbGNfcHcx

It should be noted that EODMS will be configured with SSL so clients will have to query these OGC services through HTTPS.

As mentioned above, to use the SOAP interface, users must use the WSS extension.
Example:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Header>
<wsse:Security soapenv:mustUnderstand="1" xmlns:wsse="http://docs.oasisopen.
org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
<wsse:UsernameToken xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-wssecurity-utility-1.0.xsd">
<wsse:Username>billc</wsse:Username>
<wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-username-token-profile-1.0#PasswordText">billc_pw1</wsse:Password>
</wsse:UsernameToken>
</wsse:Security>
```

```
</soapenv:Header>
<soapenv:Body>
<!-- CSW XML request goes here -->
</soapenv:Body>
</soapenv:Envelope>
```

8.2 The CSW Service Interface

The CSW Service Interface implements version 2.0.2 of the OGC CSW specification and is part of the server side component of the Meta Manager application. It can receive requests through HTTP(S) GET and/or POST as well as SOAP. It contains the following operations:

- **GetCapabilities**;
- **GetRecords**;
- **GetRecordById**; and,
- **DescribeRecord**.

The Meta Manager CSW endpoint is:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw

8.2.1 GetCapabilities

The **GetCapabilities** operation allows clients the ability to retrieve service metadata that describes the Meta Manager CSW 2.0.2 service.

The following is a **GetCapabilities** request (using HTTPS GET) example:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw?request=GetCapabilities&service=CSW&version=2.0.2

8.2.2 DescribeRecord

The **DescribeRecord** operation allows a client to discover elements of the information model supported by the Meta Manager CSW. For example, a user could ask Meta Manager to describe the element csw:Record from the base profile or MD_Metadata from the ISO profile. After this operation has been successfully invoked, it will return the XML schema definition for the element being queried. The operation allows some or all of the information model to be described.

The following is a **DescribeRecord** request example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<DescribeRecord
service="CSW"
version="2.0.2"
outputFormat="application/xml"
schemaLanguage="http://www.w3.org/2001/XMLSchema"
xmlns="http://www.opengis.net/cat/csw/2.0.2"
```

```
xmlns:csw="http://www.opengis.net/cat/csw/2.0.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/cat/csw/2.0.2
../../../csw/2.0.2/CSW-discovery.xsd">
<TypeName>csw:Record</TypeName>
</DescribeRecord>
```

8.2.3 GetRecordById

The **GetRecordById** operation allows users to retrieve products from a collection based on an identifier. After this operation has been successfully invoked, the products matching the IDs are returned.

The following is a **GetRecordById** request (using HTTP GET) example:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/eodms_catalog?request=GetRecordById&id=1515826&service=CSW&version=2.0.2

Other parameters that may be included in the **GetRecordById** request are:
• **ElementSetName**: this parameter controls the level of detail returned in the response. Possible values include:
• brief;
• summary; and,
• full.
• **outputFormat**: this parameter controls the MIME type returned for the response. Possible values include:
• application/xml; and,
• text/html.
• **outputSchema**: this parameter controls the format of the metadata in the response. Possible values include:
• http://www.opengis.net/cat/csw/2.0.2 (for Dublin Core);
• http://www.isotc211.org/2005/gmd (for ISO/NAP); and,
• urn:oasis:names:tc:ebxml-regrep:xsd:rim:3.0 (for ebRIM XML).

This is a **GetRecordById** request (using HTTP GET) example using some of these additional parameters to return the full details in NAP format.

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/eodms_catalog?request=GetRecordById&id=1515826&service=CSW&version=2.0.2&ElementSetName=full&outputSchema=http://www.isotc211.org/2005/gmd

8.2.4 GetRecords

The **GetRecords** operation allows users to search the defined data collections using the core queryables defined for each supported profile and enables results to be returned in ISO19115, NAP, Dublin Core, and ebRIM formats.

All collections defined in the EODMS Web application have been created using the North American Profile (NAP) profile. This means that they may be queried using the Dublin Core (the default profile) or the ISO 19115 queryables.

Any collection defined in EODMS can be searched via the CSW interface provided the appropriate credentials are given.

Below is the list of the available collections and their associated CSW URLs that may be searched in EODMS.

1. EODMS CSW Catalog - No special permissions are required to search this endpoint:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/eodms_catalog

2. National Air Photo Library - No special permissions are required to search this endpoint:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/eodms_napl_catalog

3. COSMO-SKYMED - Permissions are required to search this endpoint.

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/COSMO-SkyMed1

4. Disaster Monitoring Constellation - Permissions are required to search this CSW endpoint:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/DMC

5. GeoEye-1 - Permissions are required to search this endpoint:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/GeoEye-1

6. IKONOS - Permissions are required to search this endpoint.

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/IKONOS

7. Indian Remote Sensing - Permissions are required to search this endpoint.

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/IRS

~~8. NOAA - Permissions are required to search this endpoint.~~

~~https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/NOAA~~

9. QuickBird-2 - Permissions are required to search this endpoint.

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/QuickBird-2

10. RADARSAT-1 - Permissions are required to search this endpoint (will be updated to be open).

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/Radarsat1

11. RADARSAT-2 - Permissions are required to search this endpoint.

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/Radarsat2

12. RapidEye - Permissions are required to search this endpoint.

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/RapidEye

13. TERRASAR-X - Permissions are required to search this endpoint.

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/TerraSarX

14. WorldView-1 - Permissions are required to search this endpoint.

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/WorldView-1

15. WorldView-2 - Permissions are required to search this endpoint.

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/WorldView-2

The following is a **GetRecords** request example using ISO19115 queryables:

```
<csw:GetRecords service='CSW' version='2.0.2'
maxRecords='10'
startPosition='1'
resultType='results'
outputFormat='application/xml'
outputSchema='http://www.isotc211.org/2005/gmd'
xmlns='http://www.opengis.net/cat/csw/2.0.2'
xmlns:csw='http://www.opengis.net/cat/csw/2.0.2'
xmlns:ogc='http://www.opengis.net/ogc'
xmlns:ows='http://www.opengis.net/ows'
xmlns:gmd='http://www.isotc211.org/2005/gmd'
xmlns:apiso='http://www.opengis.net/cat/csw/apiso/1.0'
xmlns:dc='http://purl.org/dc/elements/1.1/'
xmlns:dct='http://purl.org/dc/terms/'
xmlns:gml='http://www.opengis.net/gml'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://www.opengis.net/cat/csw/2.0.2
http://schemas.opengis.net/csw/2.0.2/CSW-discovery.xsd'>
<csw:Query typeNames='gmd:MD_Metadata'>
<csw:ElementSetName typeNames='gmd:MD_Metadata'>full</csw:ElementSetName>
<csw:Constraint version='1.1.0'>
```

```
<ogc:Filter>
<ogc:And>
<ogc:PropertyIsLike escapeChar='\' singleChar='?' wildCard='*'>
<ogc:PropertyName>apiso:title</ogc:PropertyName>
<ogc:Literal>*imagery*</ogc:Literal>
</ogc:PropertyIsLike>
<ogc:PropertyIsLike escapeChar='\' singleChar='?' wildCard='*'>
<ogc:PropertyName>apiso:creator</ogc:PropertyName>
<ogc:Literal>*NRCan*</ogc:Literal>
</ogc:PropertyIsLike>
<ogc:BBOX>
<ogc:PropertyName>apiso:BoundingBox</ogc:PropertyName>
<gml:Envelope>
<gml:lowerCorner>-87.05 26.46</gml:lowerCorner>
<gml:upperCorner>-77.24 28.42</gml:upperCorner>
</gml:Envelope>
</ogc:BBOX>
</ogc:And>
</ogc:Filter>
</csw:Constraint>
</csw:Query>
</csw:GetRecords>
```

This is a **GetRecords** request example using Dublin Core queryables:

```
<csw:GetRecords service='CSW' version='2.0.2'
maxRecords='10'
startPosition='1'
resultType='results'
outputFormat='text/html'
outputSchema='http://www.opengis.net/cat/csw/2.0.2'
xmlns='http://www.opengis.net/cat/csw/2.0.2'
xmlns:csw='http://www.opengis.net/cat/csw/2.0.2'
xmlns:ogc='http://www.opengis.net/ogc'
xmlns:ows='http://www.opengis.net/ows'
xmlns:dc='http://purl.org/dc/elements/1.1/'
xmlns:dct='http://purl.org/dc/terms/'
xmlns:gml='http://www.opengis.net/gml'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://www.opengis.net/cat/csw/2.0.2
http://schemas.opengis.net/csw/2.0.2/CSW-discovery.xsd'>
<csw:Query typeNames='csw:Record'>
<csw:ElementSetName typeNames='csw:Record'>full</csw:ElementSetName>
<csw:Constraint version='1.1.0'>
<ogc:Filter>
<ogc:And>
<ogc:PropertyIsLike escapeChar='\' singleChar='?' wildCard='*'>
<ogc:PropertyName>dc:title</ogc:PropertyName>
<ogc:Literal>*imagery*</ogc:Literal>
</ogc:PropertyIsLike>
<ogc:PropertyIsLike escapeChar='\' singleChar='?' wildCard='*'>
<ogc:PropertyName>dc:creator</ogc:PropertyName>
<ogc:Literal>*NRCan*</ogc:Literal>
</ogc:PropertyIsLike>
<ogc:BBOX>
<ogc:PropertyName>ows:BoundingBox</ogc:PropertyName>
```

```
<gml:Envelope>
<gml:lowerCorner>-87.05 26.46</gml:lowerCorner>
<gml:upperCorner>-77.24 28.42</gml:upperCorner>
</gml:Envelope>
</ogc:BBOX>
</ogc:And>
</ogc:Filter>
</csw:Constraint>
</csw:Query>
</csw:GetRecords>
```

8.3 The WFS Service Interface

A Web Feature Service (WFS) allows a client to retrieve geospatial data, represented as features, that are encoded in Geography Markup Language (GML). The WES WFS service interface implements versions 2.0.0 and 1.1.0 of the OGC WFS specification. It can receive requests through HTTP(S) GET and/or POST. It contains the following operations:

- **GetCapabilities**;
- **DescribeFeatureType**; and
- **GetFeature**.

The WES WFS endpoint is: *https://www.eodms-sgdot.nrcan-rncan.gc.ca/GFM/services*

8.3.1 GetCapabilities

The **GetCapabilities** operation returns a response XML document describing which feature types the WFS can service, and what operations are supported on each feature type.

The following is a **GetCapabilities** request (using HTTPS GET) example:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/GFM/services?request=GetCapabilities&service=WFS&version=1.1.0

The following is a **GetCapabilities** request (using HTTPS POST) example:

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetCapabilities service="WFS" xmlns:wfs="http://www.opengis.net/wfs"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd"/>
```

8.3.2 DescribeFeatureType

The **DescribeFeatureType** operation returns an XML schema description of feature types offered by the WFS. The schema descriptions define how a WFS expects feature instances to be encoded on output in response to a GetFeature operation.

The following is a **DescribeFeatureType** request (using HTTPS GET) example that will return a description of all available feature types:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/GFM/services?request=DescribeFeatureType&service=WFS&version=1.1.0

The following is a **DescribeFeatureType** request (using HTTPS POST) example this will return a description of just the **Radarsat1** feature type:

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:DescribeFeatureType version="1.1.0"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:gml="http://www.opengis.net/
gml"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:xsi="http://www.w3.org/2001/
XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs
http://schemas.opengis.net/wfs/1.1.0/wfs.xsd">
<wfs:TypeName>Radarsat1</wfs:TypeName>
</wfs:DescribeFeatureType>
```

8.3.3 GetFeature

The **GetFeature** operation returns an XML document containing a selection of features. The client can constrain the query spatially and non-spatially to restrict which features are returned. Below is the list of the available collections and their associated WFS feature type names that may be searched in EODMS. See Section 4.3 for collection definitions.

1. National Air Photo Library - No special permissions are required to search this endpoint.

• WFS Feature Type Name: NAPL

2. COSMO-SKYMED - Permissions are required to search this endpoint.

• WFS Feature Type Name: COSMO-SkyMed1

3. Disaster Monitoring Constellation - Permissions are required to search this endpoint.

• WFS Feature Type Name: DMC

4. GeoEye-1 - Permissions are required to search this endpoint.

• WFS Feature Type Name: GeoEye-1

5. IKONOS - Permissions are required to search this endpoint.

• WFS Feature Type Name: IKONOS

6. Indian Remote Sensing - Permissions are required to search this endpoint.

• WFS Feature Type Name: IRS

7. NOAA - Permissions are required to search this endpoint.

• WFS Feature Type Name: NOAA

8. QuickBird-2 - Permissions are required to search this endpoint.

• WFS Feature Type Name: QuickBird-2

9. RADARSAT-1 - Permissions are required to search this endpoint.

• WFS Feature Type Name: Radarsat1

10. RADARSAT-2 - Permissions are required to search this endpoint.

• WFS Feature Type Name: Radarsat2

11. RapidEye - Permissions are required to search this endpoint.

• WFS Feature Type Name: RapidEye

12. TERRASAR-X - Permissions are required to search this endpoint.

• WFS Feature Type Name: TerraSarX

13. WorldView-1 - Permissions are required to search this endpoint.

• WFS Feature Type Name: WorldView-1

14. WorldView-2 - Permissions are required to search this endpoint.

• WFS Feature Type Name: WorldView-2

To determine which fields can be used to constrain results, a **DescribeFeatureType** operation can be performed on a given feature type. The list of elements of the feature will be both the search and the result fields for that feature type.

The following is a **GetFeature** request example that constrains the **WorldView-1** feature to results that have a **sequence_id** with a value of **5036606**.

```
<?xml version="1.0" ?>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="text/xml;
subtype=gml/3.1.1"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:ogc="http://www.opengis.net/
ogc"
xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:xsi="http://www.w3.org/
2001/XMLSchema-instance"
```

```
xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://
schemas.opengis.net/wfs/1.1.0/wfs.xsd"
maxFeatures="10">
<wfs:Query typeName="WorldView-1">
<ogc:Filter>
<ogc:PropertyIsEqualTo>
<ogc:PropertyName>sequence_id</ogc:PropertyName>
<ogc:Literal>5036606</ogc:Literal>
</ogc:PropertyIsEqualTo>
</ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

The following is a **GetFeature** request example that constrains the **WorldView-2** feature to results that have a **creation_date** between **2010-01-01** and **2011-12-31** and a **footprint** that intersects with the a search box defined by the lower corner coordinates of **(50, -155)** and the upper corner coordinates of **(75, -100)**.

```
<?xml version="1.0" ?>
<wfs:GetFeature service="WFS" version="1.1.0" outputFormat="text/xml;
subtype=gml/3.1.1"
xmlns:wfs="http://www.opengis.net/wfs" xmlns:ogc="http://www.opengis.net/
ogc"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wfs/2.0 http://
schemas.opengis.net/wfs/1.1.0/wfs.xsd"
maxFeatures="10">
<wfs:Query typeName="WorldView-2">
<ogc:Filter>
<ogc:And>
<ogc:PropertyIsBetween>
<ogc:PropertyName>creation_date</ogc:PropertyName>
<ogc:LowerBoundary>
<ogc:Literal>2010-01-01T00:00:00.00</ogc:Literal>
</ogc:LowerBoundary>
<ogc:UpperBoundary>
<ogc:Literal>2011-12-31T23:59:59.99</ogc:Literal>
</ogc:UpperBoundary>
</ogc:PropertyIsBetween>
<ogc:BBOX>
<ogc:PropertyName>footprint</ogc:PropertyName>
<gml:Envelope xmlns:gml="http://www.opengis.net/gml"
srsName="urn:ogc:def:crs:EPSG::4326">
<gml:lowerCorner>50 -155</gml:lowerCorner>
<gml:upperCorner>75 -100</gml:upperCorner>
</gml:Envelope>
</ogc:BBOX>
</ogc:And>
</ogc:Filter>
</wfs:Query>
</wfs:GetFeature>
```

8.4 The WCS Service Interface

The WES OGC WCS service is used to order products from EODMS through an OGC WCS

interface. The service provides the ability to query for optional parameters that may be used to modify the final product. The WES OGC WCS service receives product order requests through HTTPS GET and/or POST and has support for clients that subscribe to OGC WCS versions 2.0.1, 2.0.0, 1.1.2, and 1.1.1. To order products, users must specify the collection name as well as the identifier of the product being ordered.

The basic flow for using the WES OGC WCS service for ordering involves the following:

1. A client uses the CSW interface to discover a product of interest.
2. The client uses the WCS service operation **DescribeCoverage** to return the order parameters for that product.
3. The client uses the WCS service operation **GetCoverage** to order the product.

Section 8.4.4 contains a full end-to-end sample use case for ordering from EODMS via the WCS. The WES OGC WCS service interface contains the following operations:

- **GetCapabilities**;
- **GetCoverage**; and,
- **DescribeCoverage**.

The WES OGC WCS endpoint URL is:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/wes/services/WESOrder/wcs

8.4.1 GetCapabilites

The **GetCapabilities** operation allows a client to request information about the server's capabilities and coverages offered.

The following is a **GetCapabilities** request example, version 2.0.X:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/wes/services/WESOrder/wcs?SERVICE=WCS&VERSION=2.0.1&REQUEST=GetCapabilities

The following is a **GetCapabilities** request example, version 1.1.X:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/wes/services/WESOrder/wcs?SERVICE=WCS&VERSION=1.1.2&REQUEST=GetCapabilities

8.4.2 DescribeCoverage

The **DescribeCoverage** operation provides two main functions within EODMS:

• It provides a way for clients to retrieve the order parameters for a particular product. To do this, clients provide the collection name and the identifier of the product they are interested in ordering.

• It allows a client to discover elements of the information model supported by the WES Search service. After this operation has been successfully invoked, it will return the XML schema definition for the element being queried. The operation allows some or all of the information model to be described.

The following is an example of a **DescribeCoverage** GET request using WCS version 2.0.X which specifies a coverageId and CollectionId:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/wes/services/WESOrder/wcs?SERVICE=WCS&version=2.0.1&REQUEST=DescribeCoverage&coverageId=1519826&CollectionId=Products

The following is an example of a **DescribeCoverage** GET request using WCS version 1.1.X which specifies an identifier (this is a combination of <COLLECTIONNAME>:<RECORDID>):

https://www.eodms-sgdot.nrcan-rncan.gc.ca/wes/services/WESOrder/wcs?SERVICE=WCS&version=1.1.2&REQUEST=DescribeCoverage&identifier=Products:f31e1956-82fc-4717-bd4c-20883a1a10aa-3110587168

8.4.3 GetCoverage

The **GetCoverage** operation allows a user to submit an order to the WES Order Service via a WCS service interface. Users specify the collection ID, product ID, destination, and the preferred order parameter options for the product they are ordering.

GetCoverage version 2.0.X supports the optional *destination* parameter which can support streaming, FTP, or physical delivery of an order.

The following is an example of a **GetCoverage** GET request using WCS version 2.0.X. It also includes the optional *destination* parameter:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/wes/services/WESOrder/wcs?SERVICE=WCS&version=2.0.1&REQUEST=GetCoverage&coverageId=<COVERAGEID>&CollectionId=<COLLECTIONNAME>&format=applicationgml+xml&parameters=File%20Format(JPG),BitsPerPixel(1,4,8)&destination=type(FTP),host(ftp.compusult.net),path(/path),username(user),password(pass),filename(testing.jpg)

The *destination* parameter may be formatted in one of the following ways:

destination=type(FTP),host(<HOSTNAME>),path(<PATH>),username(<USERNAME>),password(<PASSWORD>),filename(<FILENAME>)
destination=type(PHYSICAL),Address1(ADDRESS1),Address2(ADDRESS2),Address3(ADDRES

S3),City(CITY),Classification(CLASSIFICATION),Country(COUNTRY),CustomerName(CUSTOMERNAME),EmailAddress(EMAILADDRESS),LocationName(LOCATIONNAME),PhoneNumber(PHONENUMBER),StateProvince(STATEPROVINCE),DeliveryMedia(DELIVERYMEDIA)

The following is an example of a **GetCoverage** GET request using WCS version 1.1.X. It also shows the optional *store* parameter set to true. Using this parameter tells the service to store the finished product on the server once it has been processed. A URL is then returned to the caller to retrieve the product.

https://www.eodms-sgdot.nrcan-rncan.gc.ca/wes/services/WESOrder/wcs?service=WCS&version=1.1.2&request=GetCoverage&identifier=<COLLECTIONNAME>:<COVERAGEID>&format=application/gml+xmlFormat:PNG;BitsPerPixel:0;Compression:ASIS&store=true

8.4.4 An End-to-End Sample

Below is an end-to-end sample use case that describes the flow of operations for ordering from NGDS via the WCS service interface. It uses CSW v2.0.2 as well as WCS v2.0.X formatted queries. User authentication, which is explained in Section 8.1, is required for each request below.

First, a user creates a CSW **GetRecords** query and submits it to the Meta Manager CSW. The URL being used in this example is:

https://www.eodms-sgdot.nrcan-rncan.gc.ca/MetaManagerCSW/csw/Radarsat2

The following XML fragment is a sample **GetRecords** query:

```
<csw:GetRecords service='CSW' version='2.0.2'
maxRecords='10'
startPosition='1'
resultType='results'
outputFormat='application/xml'
outputSchema='http://www.isotc211.org/2005/gmd'
xmlns='http://www.opengis.net/cat/csw/2.0.2'
xmlns:csw='http://www.opengis.net/cat/csw/2.0.2'
xmlns:ogc='http://www.opengis.net/ogc'
xmlns:ows='http://www.opengis.net/ows'
xmlns:gmd='http://www.isotc211.org/2005/gmd'
xmlns:apiso='http://www.opengis.net/cat/csw/apiso/1.0'
xmlns:dc='http://purl.org/dc/elements/1.1/'
xmlns:dct='http://purl.org/dc/terms/'
xmlns:gml='http://www.opengis.net/gml'
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation='http://www.opengis.net/cat/csw/2.0.2
http://schemas.opengis.net/csw/2.0.2/CSW-discovery.xsd'>
<csw:Query typeNames='gmd:MD_Metadata'>
<csw:ElementSetName typeNames='gmd:MD_Metadata'>full</csw:ElementSetName>
<csw:Constraint version='1.1.0'>
<ogc:Filter>
<ogc:And>
```

```
<ogc:PropertyIsLike escapeChar='\' singleChar='?' wildCard='*'>
<ogc:PropertyName>apiso:AnyText</ogc:PropertyName>
<ogc:Literal>*RADAR*</ogc:Literal>
</ogc:PropertyIsLike>
<ogc:BBOX>
<ogc:PropertyName>apiso:BoundingBox</ogc:PropertyName>
<gml:Envelope>
<gml:lowerCorner>-78.3 43.9</gml:lowerCorner>
<gml:upperCorner>-73.8 46.5</gml:upperCorner>
</gml:Envelope>
</ogc:BBOX>
</ogc:And>
</ogc:Filter>
</csw:Constraint>
</csw:Query>
</csw:GetRecords>
```

Next, the user processes the response and pulls out the record ID and associated collection name of the record they are interested in ordering. These pieces of information can be found in the *gmd:fileIdentifier* tag of the ISO profile. Using the XML response below, this value is equal to: Radarsat2:4532789.

The following XML fragment is a sample of a **GetRecordsResponse**:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetRecordsResponse xmlns="http://www.opengis.net/cat/csw/2.0.2">
<SearchStatus timestamp="2013-12-13T10:14:03-03:30" />
<SearchResults numberOfRecordsMatched="290"
numberOfRecordsReturned="10" nextRecord="11">
<gmd:MD_Metadata xmlns:gmd="http://www.isotc211.org/2005/gmd">
<gmd:fileIdentifier>
<gco:CharacterString xmlns:gco="http://
www.isotc211.org/2005/gco">4532789
</gco:CharacterString>
</gmd:fileIdentifier>
<gmd:language>
<gco:CharacterString xmlns:gco="http://
www.isotc211.org/2005/gco">eng
</gco:CharacterString>
</gmd:language>
<gmd:characterSet>
<gmd:MD_CharacterSetCode
codeList="http://www.isotc211.org/2005/
resources/Codelist/gmxCodelists.xml#MD_CharacterSetCode"
codeListValue="utf8">utf8</
gmd:MD_CharacterSetCode>
</gmd:characterSet>
<gmd:hierarchyLevel>
<gmd:MD_ScopeCode
codeList="http://www.isotc211.org/2005/
resources/Codelist/gmxCodelists.xml#MD_ScopeCode"
codeListValue="dataset">dataset</
gmd:MD_ScopeCode>
</gmd:hierarchyLevel>
<gmd:hierarchyLevelName>
```

```
<gco:CharacterString xmlns:gco="http://
www.isotc211.org/2005/gco">Satellite:
RADARSAT-2 Sensor: SAR
</gco:CharacterString>
</gmd:hierarchyLevelName>
<gmd:contact>
<gmd:CI_ResponsibleParty>
<gmd:organisationName>
<gco:CharacterString xmlns:gco="http://
www.isotc211.org/2005/gco">Data
Dissemination Division, Mapping
Information Branch, Earth
Sciences Sector, Natural Resources
Canada
</gco:CharacterString>
</gmd:organisationName>
</gmd:CI_ResponsibleParty>
</gmd:contact>
.
.
.
</gmd:MD_Metadata>
</SearchResults>
</GetRecordsResponse>
```

Next, the user queries the WCS using the **DescribeCoverage** operation to get the list of order parameters for the product returned above. Notice how the **wcs:CoverageId** is formatted as: Collection Name--Record ID

The following XML fragment is a sample of a **DescribeCoverage** query:

```
<wcs:DescribeCoverage
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation="http://www.opengis.net/wcs/2.0
http://schemas.opengis.net/wcs/2.0/wcsAll.xsd"
xmlns="http://www.opengis.net/wcs/2.0"
xmlns:wcs="http://www.opengis.net/wcs/2.0"
xmlns:wes="http://schema.compusult.net/services/2.6.1/WESOrder/wcs"
service="WCS"
version="2.0.1">
<wcs:CoverageId>Radarsat2--4532789</wcs:CoverageId>
</wcs:DescribeCoverage>
```

Next, the user processes the response from the WCS **DescribeCoverage** to pull out the order parameters for the product. This information can be found inside the *ServiceParameters* tag of the **CoverageDescriptions** XML.

The following XML fragment is a sample of a **CoverageDescriptions** response:

```
<?xml version="1.0" encoding="UTF-8"?>
<wcs:CoverageDescriptions xmlns:wcs="http://www.opengis.net/wcs/2.0">
<wcs:CoverageDescription xmlns:gml="http://www.opengis.net/gml/3.2"
gml:id="C4136a003-5e20-4f1f-8990-59b7b7cc4151">
<wcs:CoverageId>4532789</wcs:CoverageId>
```

<metadata xmlns="http://www.opengis.net/gmlcov/1.0"
xmlns:xlink="http://www.w3.org/1999/xlink"
xlink:href="http://www.eodms-sgdot.nrcan-
rncan.gc.ca/wes/services/WESSearch/csw/
Radarsat2?request=GetRecordById&Id=4532789&version=2.0.2&outputSc
hema=http://schema.compusult.net/services/2.2.0/WESSearch/csw&service=CSW
"
xlink:title="CSW GetRecordById" xlink:type="simple" />
<gml:domainSet>
<gml:Polygon gml:id="C2d1924fc-78c1-457c-b6b1-5e18058b2ca1"
srsDimension="2" srsName="EPSG:4326" xmlns:gml="http:/
/www.opengis.net/gml/3.2">
<gml:exterior>
<gml:LinearRing>
<gml:posList>-76.086336 45.467035 -
76.000286 45.207 -75.692916
45.255902 -75.777385 45.516058 -
76.086336 45.467035</gml:posList>
</gml:LinearRing>
</gml:exterior>
</gml:Polygon>
</gml:domainSet>
<rangeType xmlns="http://www.opengis.net/gmlcov/1.0">
<DataRecord xmlns="http://www.opengis.net/swe/2.0">
<field name="fileSize">
<Quantity>
<label>Total File Size</label>
<uom code="b" />
<value>185597952</value>
</Quantity>
</field>
</DataRecord>
</rangeType>
<wcs:ServiceParameters>
<wcs:CoverageSubtype>GridCoverage</wcs:CoverageSubtype>
<wcs:nativeFormat />
<wcs:Extension>
<DestinationTypes
xmlns="http://schema.compusult.net/services/
2.6.1/WESOrder/wcs">
<Type>FTP</Type>
<Type>Download</Type>
</DestinationTypes>
<Parameter xmlns="http://schema.compusult.net/
services/2.6.1/WESOrder">
<Name>packagingFormat</Name>
<Title>Packaging Format</Title>
<Description>Packaging Format</Description>
<Default>7z</Default>
<Choices>
<Value>7z</Value>
<Display>7-Zip (AES-256 encryption)</
Display>
<Description>7-Zip (AES-256 encryption)</
Description>
<DependentParameters />
</Choices>

```
</Parameter>
</wcs:Extension>
</wcs:ServiceParameters>
</wcs:CoverageDescription>
</wcs:CoverageDescriptions>
```

Next, the user queries the WCS again, this time using the **GetCoverage** operation to submit the order. In this request, the user specifies the order parameter values from the **wcs:CoverageDescriptions** response as well as the final destination of the ordered product.

The following XML fragment is a sample of a **GetCoverage** WCS operation:

```
<wcs:GetCoverage
xmlns:wos="http://schema.compusult.net/services/2.6.1/WESOrder"
xmlns:xsi='http://www.w3.org/2001/XMLSchema-instance'
xsi:schemaLocation="http://www.opengis.net/wcs/2.0
http://schemas.opengis.net/wcs/2.0/wcsAll.xsd"
xmlns="http://www.opengis.net/wcs/2.0"
xmlns:wcs="http://www.opengis.net/wcs/2.0"
xmlns:wes="http://schema.compusult.net/services/2.6.1/WESOrder/wcs"
service="WCS" version="2.0.1">
<wcs:Extension>
<wes:Parameters>
<wes:ParameterValue name="packagingFormat">7z</wes:ParameterValue>
</wes:Parameters>
<wes:NotificationEmail>email@nrcan.com</wes:NotificationEmail>
<wos:Destination>
<wos:FTPDestination>
<wos:Hostname>ftp.compusult.net</wos:Hostname>
<wos:Path>/path</wos:Path>
<wos:Filename>testing.jpg</wos:Filename>
<wos:Username>user</wos:Username>
<wos:Password>pass</wos:Password>
</wos:FTPDestination>
</wos:Destination>
</wcs:Extension>
<wcs:CoverageId>Radarsat2--4532789</wcs:CoverageId>
<wcs:format>application/gml+xml</wcs:format>
</wcs:GetCoverage>
```

When an FTP or physical destination type is specified, the WCS service will respond with a **GridCoverage** XML response as soon as it submits the order to the WES Order Service. The order will be delivered once it has been processed.

The following XML fragment is a sample of a **GridCoverage** response:

```
<?xml version="1.0" encoding="UTF-8"?>
<gmlcov:GridCoverage xmlns:gmlcov="http://www.opengis.net/gmlcov/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2" gml:id="a01b37093-81aa-4454-
9c0c-ea982e555de7">
<gml:description>Order #189, Item #229</gml:description>
<gml:identifier codeSpace="#a01b37093-81aa-4454-9c0cea982e555de7">
4532789</gml:identifier>
<gml:boundedBy>
```

<gml:Envelope srsName="EPSG:4326">
<gml:lowerCorner>-76.086336 45.207</gml:lowerCorner>
<gml:upperCorner>-75.692916 45.516058</gml:upperCorner>
</gml:Envelope>
</gml:boundedBy>
<gml:domainSet />
<gml:rangeSet>
<gml:File>
<gml:rangeParameters />
<gml:fileReference />
<gml:fileStructure />
</gml:File>
</gml:rangeSet>
<gmlcov:rangeType>
<swe:DataRecord xmlns:swe="http://www.opengis.net/swe/2.0">
<swe:field name="OrderId">
<swe:Text>
<swe:value>189</swe:value>
</swe:Text>
</swe:field>
<swe:field name="OrderItemId">
<swe:Text>
<swe:value>229</swe:value>
</swe:Text>
</swe:field>
</swe:DataRecord>
</gmlcov:rangeType>
</gmlcov:GridCoverage>