NNEW

NEXTGEN NETWORK ENABLED WEATHER

FY10 Weather Data Cube Capability Evaluation

Test and Evaluation Procedures

September 13, 2010

Table of Contents

Revision History

Version	Date	Contents	Editors	Contributors
0.1	6/30/2010	Draft 1 Outline	Michael Leon MarySue Schultz	Michael Leon MarySue Schultz
0.2	7/23/2010	Draft 2 Addition of WFSRI, RegRep, WCSRI and NCAR Data Retrieval	Michael Leon MarySue Schultz	Michael Leon Ai-Hoa Sanh MarySue Schultz Sudha Verma
0.3	8/20/2010	Draft 3 Addition of MDL and Service Adapters	MarySue Schultz	Don Groot Steve Olson
1.0	8/31/2010	Reformatted to support CE test order; addition of GSD Data Retrieval; edits to Service Adapters	MarySue Schultz	Don Groot MarySue Schultz
1.1	9/02/2010	Modifications to WFS Functional Testing	MarySue Schultz	MarySue Schultz
1.2	9/07/2010	Final version of RegRep procedures; final version of WCSRI procedures	MarySue Schultz	Brett Levasseur Rob Weingruber
1.3	9/08/2010	Final version of SA procedures; NEVS procedures; WFS functional procedures	MarySue Schultz	Don Groot Jennifer Mahoney
1.4	9/8/2010	Reformatting	MarySue Schultz	
1.5	9/10/2010	SA modifications; NEVS additions; WCSRI modifications	MarySue Schultz	Don Groot Paul Hamer Jen Lin
1.6	9/13/2010	WFS modifications	MarySue Schultz	Jen Lin Ai Hoa Sanh
1.8	9/13/2010	Added NWP procedures	MarySue Schultz	Melanie Flavin

1 Introduction

The purpose of the 4-D Weather Data Cube Capability Evaluation (Capability Evaluation) is to show progress made towards the development of the 4-D Weather Data Cube (Cube). The Capability Evaluation will be composed of two major activities:

1. High-Level Capability Evaluation Presentation. The presentation will include a discussion of the Cube (functionality, standards, and anticipated operational architecture), and a visual presentation that shows progress towards development of the Cube.
2. Capability Test and Evaluation (T&E). The purpose of the T&E is to thoroughly test the capabilities that have been implemented to date, and to provide performance and latency measurements.

The FY10 4-D Weather Data Cube Capability Evaluation Plan (Plan) provides an overview, including goals and objectives, and contains background information on the development of Cube functionality. The Test And Evaluation Procedures document contains the details of the T&E portion of the Capability Evaluation, and includes the test procedures that will be executed during the T&E.

1.1 Testing Location

The T&E will be conducted from the William J Hughes Technical Center (WJHTC). The test procedures and testing tools will be installed there, and WJHTC staff will execute the tests and record the results. The systems under test will reside at various locations, to simulate the distributed operational architecture as much as possible. Instances of the WCSRI and WFSRI will be located at NCAR, MIT/LL and NOAA data provider sites, and the WJHTC will host RIs to serve FAA data sets. Instances of the Reg/Rep will be located at multiple NOAA and FAA sites. Architectural diagrams of the test systems and their network connections are provided in the Plan.

1.2 Testing Process

1. A number of organizations are contributing to the FY10 T&E. To help prepare everyone for the T&E in September, three dry runs are scheduled during July and August. The purpose of the dry runs is to execute the test procedures in advance of the T&E, to test the network connections between the WJHTC and the data providers, verify that the RI and Reg/Rep capabilities work properly, and familiarize the WJHTC staff with the test procedures.

2. WJHTC staff will execute the test procedures and record the results. Each test procedure will provide a description of the test, the requirements under test, and instructions for running the test. During the T&E, the test procedures will be displayed, and an explanation of each test will be provided as the test is executed.

3. Test results will be collected and presented in two reports:

    ◦ Quick-Look Report, which provides a high-level summary of the test results.

◦ FY10 Test and Evaluation Results and Analysis document, which provides a more complete report of the test results.

1.3 System Requirements

System Dependencies

- Java 6.0
- JDK 1.6
- FUSE ESB 4.2.0-fuse-01-00
- NetCDF 4 C library
- NetCDF 4 Java Native Interface (JNI) C bindings
- HDF5 C library
- Mozilla Firefox (or similar web browser)
- Web Feature Service Reference Implementation (WFSRI) 2.0, 2.1.1
- Web Coverage Service Reference Implementation (WCSRI) 2.1

Demonstration Applications

- **Soap Client 2.0**

The WFSRI Version 2.0 provides a general purpose SOAP Client, for sending SOAP requests to the RIs.

- **WFSRI Test Project**

The request XMLs for a portion of the WFSRI tests are bundled as part of the WFSRI Test project.

- **NNEW Testing Portal**

The NNEW Testing Portal is a web application developed by GSD to facilitate running the WFS functional tests. The tests can be launched by authorized users from any web browser.

- **soapUI 3.5**

soapUI is a Open Source tool for functional testing, mainly of Web Services.

- **CubeView**

An interactive thick-client Java Web Start display for visualizing a comprehensive set of FY10 data products. It also includes the Dashboard function for visualizing the FY10 capability evaluation architecture.

- **WxConsumer** 3.4

- **Registry Administration UI**

A UI that enables publishing and discovery of datasets, service instances, service interfaces, taxonomies, and more. Its discovery features allow for searching through individual registries with local searches, or through multiple-registry federations with federated searches.

- **Registry Test Client**

A command line program used to test the fault-tolerance and client-side support for federated queries within the regrep4-client API library.

- **System Wide Information Simulator (SWIS)**

This tool is used for RASP Service Adapter testing. It will simulate LDD, METAR, and OMO data that the RASP will receive as input.

- **RASP SA Client Test Tool**

This tool is used for RASP Service Adapter testing. It pulls LDD, METAR, and OMO data from the 4-D Wx Data Cube.

- **RASP SA Data Comparison Tool**

This tool will be used to compare the LDD and METAR data archived by the RASP to the data that the RASP SA published to the 4-D Wx Data Cube.

- **NetCDF Publishing Tool**

This tool will be used in testing DOTS and ATOP Service Adapters. It publishes gridded data to the 4-D Wx Data Cube in the NetCDF4 format.

- **Integrated Data Viewer (IDV)**

The IDV is a Java-based software framework for analyzing and visualizing geoscience data. The IDV is developed at the Unidata Program Center (UPC), part of the University Corporation for Atmospheric Research (UCAR), Boulder, Colorado.

- **testWcsWx**

A testing tool for accessing gridded data from the WCSRI. The tool prompts testers for a URL and coverage, employs the WxConsumer client to handle communications with the WCSRI, and uses the IDV for data display.

2 Implementation Verification – Web Feature Service Reference Implementation (WFSRI)

2.1 Test Environment and Setup

2.1.1 WFSRI Server

The tests that follow in this chapter assume that a WFSRI Server is up and running at MIT Lincoln Laboratory, at the following endpoint:

- http://nnew-wfs1.wx.ll.mit.edu:80/wfsri-2.0/wfs

If for any reason you need to install a WFSRI Server, refer to the WFSRI 2.0 Installation Guide, available on the NNEW Wiki:

- https://wiki.ucar.edu/display/NNEWD/WFSRI+2.0+Installation

2.1.2 WFSRI Repeater

The WFSRI Repeater is an add-on to the WFSRI, providing repeater/forwarding service between two WFS servers: an Origin Server and a Distribution Server. The repeater subscribes to updates of a single Feature Type on the Origin Server and propagates the updates, as it receives them, to the Distribution Server. The WFSRI Repeater executable jar is bundled with the WFSRI 2.0 release. It is located in:

```
$WFSRI_INSTALL/repeater
```

The WFSRI Repeater requires JDK 1.6 to run. It must be run in tandem with the Distribution Server, started separately from the terminal using the following command:

```
java -jar wfsri-repeater-2.0.jar -f <file>
```

The WFSRI Repeater should be configured as outlined in the User Guide:

- https://wiki.ucar.edu/display/NNEWD/WFSRI+2.0+User+Guide#WFSRI2.0UserGuide-RunningtheWFSRIRepeaterApplication

2.1.3 SOAP Client

The WFSRI Version 2.0 provides a general purpose **SOAP Client**, which will be used for the majority of these tests. The SOAP Client can be downloaded from:

- https://wiki.ucar.edu/download/attachments/51353793/ soapclient-2.0.jar?version=1&modificationDate=1281039653250

The Soap Client requires JDK 1.6 to run. It may be executed from the terminal with the following command, replacing the placeholders as directed in the table below:

```
java -jar soapclient-2.0.jar <url> <file> <user> <pass>
```

Placeholder	Replace With
`<url>`	The SOAP URL on which the WFSRI is listening (e.g., `http://<WFSRIserver>/soap/wfs`). This is the value of `wfs.url`, set in the `wfsri.properties` file during configuration of the WFSRI.
`<file>`	The XML file containing the operation's request.
`<user>`	The appropriate Producer or consumer username (consumers have read-only access to a WFSRI, Producers have both read and write access). This would have been set in the CamelContext.xml file when configuring the WFSRI
`<pass>`	The appropriate Producer or consumer password (consumers have read-only access to a WFSRI, Producers have both read and write access). Like the username, this would have been set in the CamelContext.xml file when configuring the WFSRI

2.1.4 Request XMLs

The request XMLs for the outlined tests are bundled as part of the WFSRI Test project. To download the XMLs, checkout the WFSRI Test project from wxforge using :

```
svn co http://wxforge.wx.ll.mit.edu/svn/wfsri-test/tags/wfsri-test-2.0/ wfsri-test
```

The XML files can be found in the `src/test/resources/xml` folder.

2.2 GetCapabilities

To retrieve the capabilities offered by the WFS server, you can pass in a GetCapabilities request to the Soap Client. A GetCapabilities request is included in the WFSRI Test project.

( ) 1. Type the following command into the terminal and compare the response to the highlighted code below:

```
cat wfs-requests/getCapabilities.xml
```

```
<wfs:GetCapabilities service="WFS" xmlns:wfs="http://www.opengis.net/wfs/2.0">
  <ows:Sections xmlns:ows="http://www.opengis.net/ows/1.1">
    <ows:Section>ServiceIdentification</ows:Section>
    <ows:Section>ServiceProvider</ows:Section>
    <ows:Section>OperationsMetadata</ows:Section>
  </ows:Sections>
</wfs:GetCapabilities>
```

( ) 2. Run the following command to use the SOAP Client to execute the GetCapabilities operation on the Origin Server:

```
java -jar soapclient-2.0.jar http://nnew-wfs1.wx.ll.mit.edu/wfs-2.0/wfs wfs-
    requests/getCapabilities.xml <User> <Password>
```

( ) 3. Verify that this command returns the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:WFS_Capabilities xmlns:ows="http://www.opengis.net/ows/1.1"
 xmlns:ogc="http://www.opengis.net/ogc" xmlns:wfs="http://www.opengis.net/wfs/2.0"
 xmlns:gml="http://www.opengis.net/gml/3.2"
 xmlns:xlink="http://www.w3.org/1999/xlink"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xmlns:fes="http://www.opengis.net/fes/2.0"
 xsi:schemaLocation="http://www.opengis.net/wfs/2.0 ../../../../../../ogc-
  bindings/schemas/net/opengis/wfs/2.0.0/wfs.xsd"
 version="2.0.0" updateSequence="0">

 <ows:ServiceIdentification>
  <ows:Title>MIT/LL WFS</ows:Title>
  <ows:Abstract>Web Feature Service maintained by MIT/LL, serving NNEW data
   providers; contact claypool@ll.mit.edu</ows:Abstract>
  <ows:Keywords>
   <ows:Keyword>MIT/LL</ows:Keyword>
   <ows:Keyword>NNEW</ows:Keyword>
   <ows:Keyword>MIT Lincoln Laboratory</ows:Keyword>
   <ows:Type>String</ows:Type>
  </ows:Keywords>
  <ows:ServiceType>WFS</ows:ServiceType>
  <ows:ServiceTypeVersion>2.0.0</ows:ServiceTypeVersion>
  <ows:Fees>None</ows:Fees>
  <ows:AccessConstraints>None</ows:AccessConstraints>
 </ows:ServiceIdentification>

 <ows:ServiceProvider>
  <ows:ProviderName>MIT Lincoln Laboratory</ows:ProviderName>
  <ows:ProviderSite xlink:href="http://www.ll.mit.edu/"/>
  <ows:ServiceContact>
   <ows:IndividualName>Kajal Claypool</ows:IndividualName>
   <ows:PositionName>Technical Staff</ows:PositionName>
   <ows:ContactInfo>
    <ows:Phone>
     <ows:Voice>781.981.5404</ows:Voice>
     <ows:Facsimile>781.981.WOOD</ows:Facsimile>
    </ows:Phone>
    <ows:Address>
     <ows:DeliveryPoint>244 Wood Street</ows:DeliveryPoint>
     <ows:City>Lexington</ows:City>
     <ows:AdministrativeArea>Group 43</ows:AdministrativeArea>
     <ows:PostalCode>02420</ows:PostalCode>
     <ows:Country>USA</ows:Country>
     <ows:ElectronicMailAddress>claypool@ll.mit.edu</ows:ElectronicMailAddress>
    </ows:Address>
    <ows:OnlineResource xlink:href="http://wxforge.wx.ll.mit.edu:8080/jira"/>
    <ows:HoursOfService>24x7</ows:HoursOfService>
    <ows:ContactInstructions>Create a JIRA issue (Web Feature Service project) with
     normal requsts; E-mail or phone Kajal for emergency requests.
    </ows:ContactInstructions>
   </ows:ContactInfo>
   <ows:Role>PointOfContact</ows:Role>
  </ows:ServiceContact>
 </ows:ServiceProvider>

 <ows:OperationsMetadata>
  <ows:Operation name="GetCapabilities">
   <ows:DCP>
    <ows:HTTP>
     <ows:Post xlink:href="http://ngen-wfsri.wx.ll.mit.edu/wfsri-1.3/wfs"/>
    </ows:HTTP>
   </ows:DCP>
  </ows:Operation>
  <ows:Operation name="DescribeFeatureType">
   <ows:DCP>
    <ows:HTTP>
     <ows:Post xlink:href="http://ngen-wfsri.wx.ll.mit.edu/wfsri-1.3/wfs"/>
    </ows:HTTP>
   </ows:DCP>
  </ows:Operation>
  <ows:Operation name="GetFeature">
   <ows:DCP>
    <ows:HTTP>
     <ows:Post xlink:href="http://ngen-wfsri.wx.ll.mit.edu/wfsri-1.3/wfs"/>
    </ows:HTTP>
   </ows:DCP>
  </ows:Operation>
  <ows:Operation name="Transaction">
   <ows:DCP>
    <ows:HTTP>
     <ows:Post xlink:href="http://ngen-wfsri.wx.ll.mit.edu/wfsri-1.3/wfs"/>
    </ows:HTTP>
   </ows:DCP>
  </ows:Operation>
 </ows:OperationsMetadata>

 <wfs:FeatureTypeList>
   <wfs:FeatureType>
     <wfs:Name xmlns:wx="http://www.eurocontrol.int/wx/1.1">wx:MotionVector</wfs:Name>
     <wfs:Title>MotionVector</wfs:Title>
     <wfs:SRS>urn:ogc:def:crs:EPSG::4326</wfs:SRS>
   </wfs:FeatureType>
   <wfs:FeatureType>
     <wfs:Name xmlns:nawx="http://www.faa.gov/nawx/1.2">nawx:LeadingEdge</wfs:Name>
     <wfs:Title>LeadingEdge</wfs:Title>
     <wfs:SRS>urn:ogc:def:crs:EPSG::4326</wfs:SRS>
   </wfs:FeatureType>
   <wfs:FeatureType>
     <wfs:Name xmlns:nawx="http://www.faa.gov/nawx/1.2">nawx:ScoredRegion</wfs:Name>
     <wfs:Title>ForecastAccuracy</wfs:Title>
     <wfs:SRS>urn:ogc:def:crs:EPSG::4326</wfs:SRS>
   </wfs:FeatureType>
   <wfs:FeatureType>
     <wfs:Name xmlns:nawx="http://www.faa.gov/nawx/1.2">nawx:EchoTopPoint</wfs:Name>
     <wfs:Title>EchoTopTag</wfs:Title>
     <wfs:SRS>urn:ogc:def:crs:EPSG::4326</wfs:SRS>
   </wfs:FeatureType>
   <wfs:FeatureType>
     <wfs:Name xmlns:wx="http://www.eurocontrol.int/wx/1.1">wx:Contour</wfs:Name>
     <wfs:Title>Contour</wfs:Title>
     <wfs:SRS>urn:ogc:def:crs:EPSG::4326</wfs:SRS>
   </wfs:FeatureType>
   <wfs:FeatureType
     xsi:schemaLocation="http://www.opengis.net/wfs/2.0 ../../../../../../../ogc-
bindings/schemas/net/opengis/wfs/2.0.0/wfs.xsd">
     <wfs:Name xmlns:avwx="http://www.eurocontrol.int/avwx/1.1">avwx:TAF</wfs:Name>
     <wfs:Title>TAF</wfs:Title>
     <wfs:DefaultCRS>urn:ogc:def:crs:EPSG::4326</wfs:DefaultCRS>
   </wfs:FeatureType>
 </wfs:FeatureTypeList>

 <fes:Filter_Capabilities>
  <fes:Id_Capabilities>
   <fes:ResourceIdentifier name="fes:ResourceId">
    <ows:Metadata/>
   </fes:ResourceIdentifier>
  </fes:Id_Capabilities>
  <fes:Scalar_Capabilities>
   <fes:LogicalOperators/>
   <fes:ComparisonOperators>
    <fes:ComparisonOperator name="PropertyIsEqualTo"/>
    <fes:ComparisonOperator name="PropertyIsNotEqualTo"/>
    <fes:ComparisonOperator name="PropertyIsLessThan"/>
    <fes:ComparisonOperator name="PropertyIsGreaterThan"/>
    <fes:ComparisonOperator name="PropertyIsLessThanOrEqualTo"/>
```

```
      <fes:ComparisonOperator name="PropertyIsGreaterThanOrEqualTo"/>
     </fes:ComparisonOperators>
    </fes:Scalar_Capabilities>
    <fes:Spatial_Capabilities>
     <fes:GeometryOperands>
      <fes:GeometryOperand name="gml:AbstractGeometricPrimitive"/>
     </fes:GeometryOperands>
     <fes:SpatialOperators>
      <fes:SpatialOperator name="BBOX"/>
      <fes:SpatialOperator name="Equals"/>
      <fes:SpatialOperator name="Disjoint"/>
      <fes:SpatialOperator name="Intersects"/>
      <fes:SpatialOperator name="Touches"/>
      <fes:SpatialOperator name="Crosses"/>
      <fes:SpatialOperator name="Within"/>
      <fes:SpatialOperator name="Contains"/>
      <fes:SpatialOperator name="Overlaps"/>
     </fes:SpatialOperators>
    </fes:Spatial_Capabilities>
   </fes:Filter_Capabilities>

  </wfs:WFS_Capabilities>
```

2.3 DescribeFeatureType Using the Generic Client

The DescribeFeatureType request returns the XML Schema descriptions of the specified Feature Type.

1. Type the following command into the terminal and compare the response to the highlighted code below:

```
cat wfsri-requests/describeContours.xml

<wfs:DescribeFeatureType
  version="2.0.0"
  service="WFS"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:wx="http://www.eurocontrol.int/wx/1.1">
  <wfs:TypeName>wx:Contour</wfs:TypeName>
</wfs:DescribeFeatureType>
```

2. Run the following command to execute the Soap Client and run the DescribeFeatureType operation on the Origin Server:

```
java -jar soapclient-2.0.jar http://nnew-wfs1.wx.ll.mit.edu/wfs-2.0/wfs wfsri-requests/describeContours.xml <User>
    <Password>
```

3. Verify that this command returns the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:DescribeFeatureTypeResponse xmlns:wfs="http://www.opengis.net/wfs-util">
  <xsd:schema xmlns=http://www.w3.org/2001/XMLSchema
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:gml=http://www.opengis.net/gml/3.2
    xmlns:wx=http://www.eurocontrol.int/wx/1.1
    targetNamespace="http://www.eurocontrol.int/wx/1.1" elementFormDefault="qualified"
    attributeFormDefault="unqualified">
    <!-- Schema auto-generated by FullMoon, applying rule suite xmi11ea -->
    <annotation>
      <documentation>[WARN-A001] - No package description in UML model
      </documentation>
    </annotation>
    <import namespace="http://www.opengis.net/gml/3.2"
      schemaLocation="../../../../net/opengis/gml/3.2.1/gml.xsd"/>
    <include schemaLocation="wxBase.xsd"/>
    <element name="Contour" substitutionGroup="wx:AbstractWxFeature" type="wx:ContourType">
      <annotation>
        <documentation>An approximated or actual outline or border of weather conditions.
        </documentation>
      </annotation>
    </element>
    <complexType name="ContourType">
      <complexContent>
        <extension base="wx:AbstractWxFeatureType">
          <sequence>
            <element maxOccurs="1" minOccurs="0" name="contourValue" type="gml:MeasureType"/>
            <element name="geometry" type="gml:CurvePropertyType"/>
          </sequence>
        </extension>
      </complexContent>
    </complexType>
    <complexType name="ContourPropertyType">
      <sequence minOccurs="0">
        <element ref="wx:Contour"/>
      </sequence>
      <attributeGroup ref="gml:AssociationAttributeGroup"/>
      <attributeGroup ref="gml:OwnershipAttributeGroup"/>
    </complexType>
  </xsd:schema>
</wfs:DescribeFeatureTypeResponse>
```

2.4 GetFeature Request/Response

The GetFeature Request/Response interface allows users to retrieve feature data from the WFS server for a specified feature type. The Soap Client can be used for retrieving unfiltered as well as filtered data from a WFS server.

2.4.1 Unfiltered Access

The file `motionVectorsAll.xml` (shown below) shows a simple OGC filter that, when executed, retrieves all MotionVector data that is available at the server.

1. Type the following command into the terminal and compare the response to the highlighted code below:

```
cat wfsri-requests/motionVectorsAll.xml

<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetFeature service="WFS" version="2.0.0" outputFormat="text/xml;
  subtype=gml/3.2.1" xmlns:nawx=http://www.faa.gov/nawx/1.2
  xmlns:wx=http://www.eurocontrol.int/wx/1.1
  xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:wfs=http://www.opengis.net/wfs/2.0
  xmlns:gml=http://www.opengis.net/gml/3.2
  xmlns:fes=http://www.opengis.net/fes/2.0">

  <wfs:Query typeNames="wx:MotionVector" />
</wfs:GetFeature>
```

2. Run the following command to execute the Soap Client and run the GetFeature operation on the Origin Server:

```
java -jar soapclient-2.0.jar http://nnew-wfs1.wx.ll.mit.edu/wfs-2.0/wfs wfs-
    requests/motionVectorsAll.xml <User> <Password>
```

3. Verify that this command returns a response similar to the following:

```
<ns:FeatureCollection numberReturned="1" timeStamp="2010-08-11T19:53:28.427Z"
  numberMatched="1" xmlns:ns="http://www.opengis.net/wfs/2.0">
  <ns:member>
    <wx:MotionVector gml:id="id5" xmlns:wfs=http://www.opengis.net/wfs/2.0
      xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
      xmlns:wx="http://www.eurocontrol.int/wx/1.1" xmlns:gml=http://www.opengis.net/gml/3.2
      xmlns:om=http://www.opengis.net/om/1.0/gml32
      xmlns:xlin="http://www.w3.org/1999/xlink" xmlns:nawx="http://www.faa.gov/nawx/1.2">

      <wx:obsOrFcstTime>
        <gml:TimeInstant gml:id="id6">
          <gml:timePosition>2008-07-09T04:00:00Z</gml:timePosition>
        </gml:TimeInstant>
      </wx:obsOrFcstTime>
      <wx:geometry>
        <gml:Point gml:id="id7" srsName="urn:ogc:def:crs:EPSG::4326" srsDimension="2">
          <gml:pos>40.3277 -83.7614</gml:pos>
        </gml:Point>
      </wx:geometry>
      <wx:speed>
        <wx:Speed uom="kt">28.66</wx:Speed>
      </wx:speed>
      <wx:direction>
        <wx:Angle uom="deg">61.12</wx:Angle>
      </wx:direction>
    </wx:MotionVector>
  </ns:member>
</ns:FeatureCollection>
```

2.4.2 Spatial Subsetting

The file `motionVectorsBBOX.xml` (shown below) shows a simple OGC filter that, when executed, retrieves all MotionVectors data that is available within the specified bounding box.

1. Type the following command into the terminal and compare the response to the highlighted code below:

```
cat wfsri-requests/echoTopTagsBBOX.xml
```

```
<?xml version="1.0" encoding="UTF-8"?>
<wfs:GetFeature service="WFS" version="2.0.0" outputFormat="text/xml;
  subtype=gml/3.2.1" xmlns:nawx=http://www.faa.gov/nawx/1.2
  xmlns:wfs=http://www.opengis.net/wfs/2.0
  xmlns:gml=http://www.opengis.net/gml/3.2
  xmlns:fes="http://www.opengis.net/fes/2.0">

  <wfs:Query typeNames="wx:MotionVector">
    <fes:Filter>
      <fes:BBOX>
        <gml:Envelope srsName="urn:ogc:def:crs:EPSG::4326" srsDimension="2">
          <gml:coordinates>19 -80 28 -61</gml:coordinates>
        </gml:Envelope>
      </fes:BBOX>
    </fes:Filter>
  </wfs:Query>
</wfs:GetFeature>
```

2. Run the following command to execute the Soap Client and run the GetFeature operation on the Origin Server:

```
java -jar soapclient-2.0.jar http://nnew-wfs1.wx.ll.mit.edu/wfs-2.0/wfs wfs-requests/motionVectorsBBOX.xml <User>
    <Password>
```

3. Verify that all results fall within the bounding box specified (19, -80, 28, -61).

2.5 WFSRI Repeater

1. Ensure the MotionVectors feature table is created on the Distribution Server by executing the following commands:

```
a. sqlplus / as sysdba
b. select table_name from all_tables where table_name='MOTIONVECTORS';
```

   c. Ensure the following is returned:

```
TABLE_NAME
------------------------------
MOTIONVECTORS
```

2. Ensure the MotionVectors feature type is registered on the Distribution Server by executing the following commands:

```
a. sqlplus / as sysdba
b. select featuretypename from RASPDIST.wfs_featuretype where
   featuretypename='MotionVector';
```

   c. Ensure the following is returned:

```
FEATURETYPENAME
------------------------------
MotionVector
```

3. Ensure the feature table is empty by executing the following commands:

```
a. sqlplus / as sysdba
b. select count(*) from CIWS_MV.MOTIONVECTORS;
```

   c. Ensure the following is returned:

```
COUNT(*)
------------------------------
0
```

   d. If the count is greater than zero, clear the feature table by executing the following command:

```
delete from CIWS_MV.MOTIONVECTORS
```

4. Type the following command into the terminal and compare the response to the highlighted code below:

```
cat /app/ll/wfsri-repeater/conf/motionVector.properties
```

```
originEndpoint=http://nnew-wfs1.wx.ll.mit.edu/wfsri-2.0/wfs
originUsername=CIWS2
originPassword=CIWS2
distributionEndpoint=http://192.168.56.9:8084/wfsri-2.0/wfs
distributionUsername=CIWS_MV
distributionPassword=ciws001##
subscribeFilename=/app/ll/wfsri-repeater/conf/motionVectorsSub.xml
```

5. Execute the WFSRI Repeater application for each CIWS Data Product Feature Type on the Distribution Server.

```
cd /app/ll/wfsri-repeater/target
```

```
java -jar wfsri-repeater-2.0.jar -f /app/ll/wfsri-repeater/conf/motionVector.properties
```

6. Verify that the feature table is populated with data on the Distribution Server by executing the following commands:

```
a. sqlplus / as sysdba
b. select count(*) from CIWS_MV.MOTIONVECTORS;
```

   c. Ensure the count is greater than zero.

3 WFSRI Version 2.0 Functional Test Procedures

3.1 Introduction

The WFSRI Version 2.0 functional tests verify that the Version 2.0 requirements have been satisfied, and that the implemented capabilities function properly.

3.2 Reference Documents

The requirements under test are listed in Table 3 of Appendix C: WCS/WFS Requirements in the 4-D Weather Data Cube FY10 Capability Evaluation Plan, version 2.0.

3.3 Test Description

3.3.1 System Under Test

The system under test consists of the NNEW WFSRI Version 2.0 and an associated Oracle Database. The WFS functional test procedures will access the WFS only, and will not directly test the database functionality.

3.3.2 Test Setup

The WFSRI Version 2.0 functional test procedures will be executed from the NNEW Testing Portal, an automated, web-enabled testing tool that has been developed at GSD for testing NextGen capabilities. The Testing Portal, the WFSRI and the Oracle database must be installed at the FAA Tech Center prior to the T&E activities. The Oracle database associated with the WFSRI must contain the sample data needed by the Portal for WFSRI 2.0 functional testing.

NNEW Testing Portal

GSD is developing a web application in-house to facilitate running the WFS functional tests. The tests can be launched by authorized users from any web browser. The tests will be based on required use cases, and may include a GetCapabilites request, GetFeatures requests, specific domains, times, etc. This web application is public facing and requires a login in order to be accessed.

The GSD NNEW Testing Portal website is split into two parts:

1. Rich Internet Application (RIA) – HTML, JavaScript and Flash deployed in Tomcat
2. Server Code – Java web services running in Tomcat and accessing MySQL

The GSD NNEW Testing Portal user chooses a test to run which invokes its RESTful web service (Server side, Testing Portal Java code, running from within Tomcat). This server side Java code then invokes the selected test against a specified WFSRI, logging the status of the request (pass or fail, plus additional diagnostic information) as well as the performance metrics, into a MySQL database. These results are then displayed on the Testing Portal user's web application interface in the form of text and charts. If the results of a test can be geocoded (that is, they can be plotted in a map), they are displayed in the Google Maps pane of the Testing Portal window. Test results are archived in a MySQL database on the server and can be queried and displayed/charted inside the Testing Portal.

3.3.3 Personnel

GSD test developers and MIT/LL engineers must be available for support (not necessarily on site).

3.3.4 Location

The WFSRI, the DB and the testing portal will all reside within the R&D Enclave at the FAA Tech Center. The tests will be executed at the Tech Center.

3.4 Test Conduct

3.4.1 Security Considerations

The WFSRI functional tests will be conducted entirely within the R&D Enclave at the FAA Tech Center. There will be no access to remote machines, and no requests or data will be transmitted over the FTI network or NOAAnet.

3.4.2 Test Data Reduction and Analysis

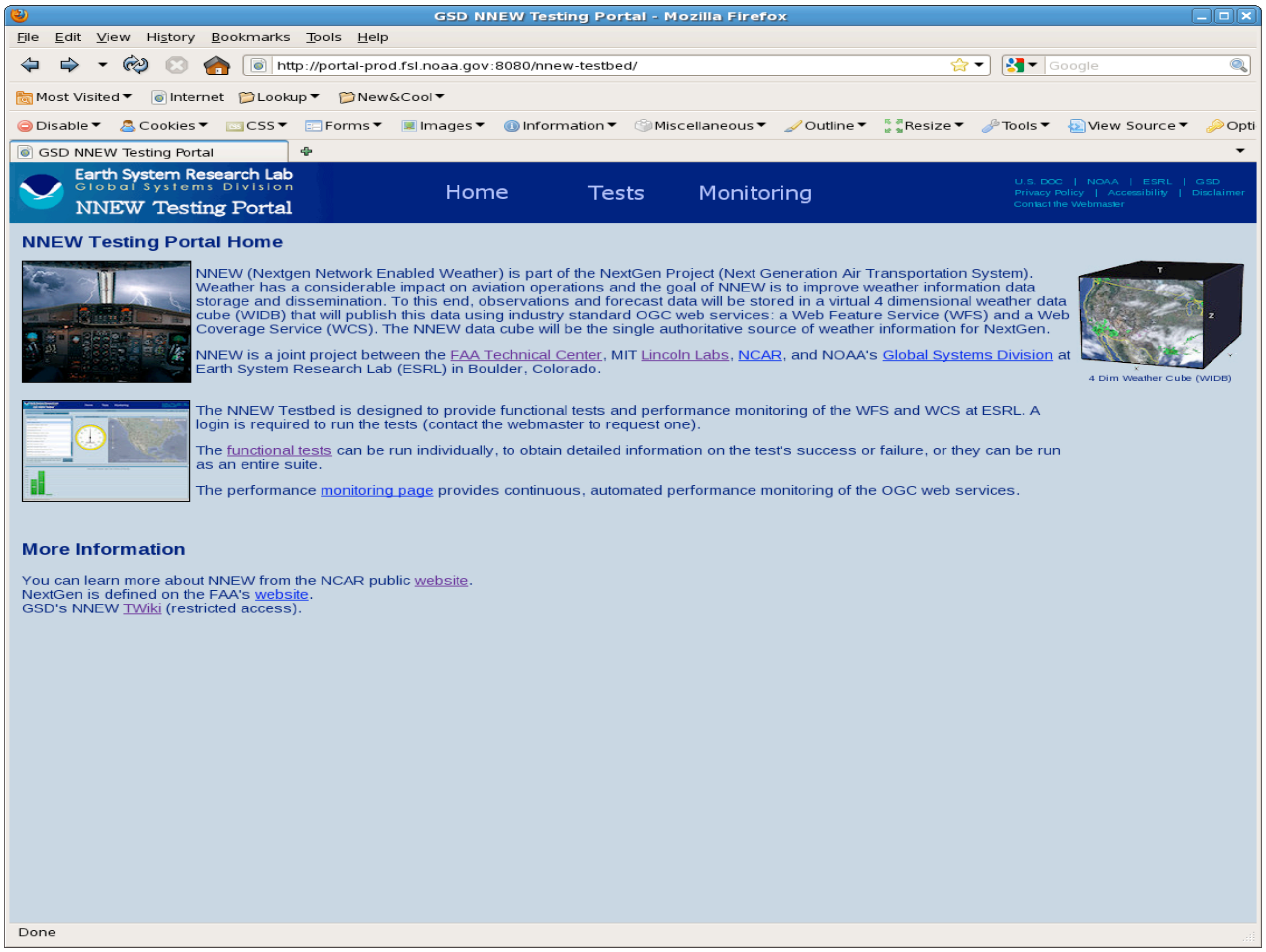
There will be two types of test results: 1) pass/fail for each test procedure; comments may also be recorded; 2) request and response times will be logged, and may be used to determine latency after the CE testing is finished.

3.4.3 Test Preparation

3.4.3.1 Testing Portal Log In

1. Bring up the NNEW Testing Portal in a web browser (preferably Mozilla Firefox) using the URL http://portal.wjhtc.nnew:8080/nnew-testbed/

Be sure to use the trailing slash. The Testing Portal Home Page is shown below:


GSD NNEW Testing Portal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://portal-prod.fsl.noaa.gov:8080/nnew-testbed/
Google
Most Visited Internet Lookup New&Cool
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Opti
GSD NNEW Testing Portal
Earth System Research Lab
Global Systems Division
NNEW Testing Portal
Home
Tests
Monitoring
U.S. DOC | NOAA | ESRL | GSD
Privacy Policy | Accessibility | Disclaimer
Contact the Webmaster
4 Dim Weather Cube (WIDB)
Done

NNEW Testing Portal Home

NNEW (Nextgen Network Enabled Weather) is part of the NextGen Project (Next Generation Air Transportation System). Weather has a considerable impact on aviation operations and the goal of NNEW is to improve weather information data storage and dissemination. To this end, observations and forecast data will be stored in a virtual 4 dimensional weather data cube (WIDB) that will publish this data using industry standard OGC web services: a Web Feature Service (WFS) and a Web Coverage Service (WCS). The NNEW data cube will be the single authoritative source of weather information for NextGen.

NNEW is a joint project between the FAA Technical Center, MIT Lincoln Labs, NCAR, and NOAA's Global Systems Division at Earth System Research Lab (ESRL) in Boulder, Colorado.

The NNEW Testbed is designed to provide functional tests and performance monitoring of the WFS and WCS at ESRL. A login is required to run the tests (contact the webmaster to request one).

The functional tests can be run individually, to obtain detailed information on the test's success or failure, or they can be run as an entire suite.

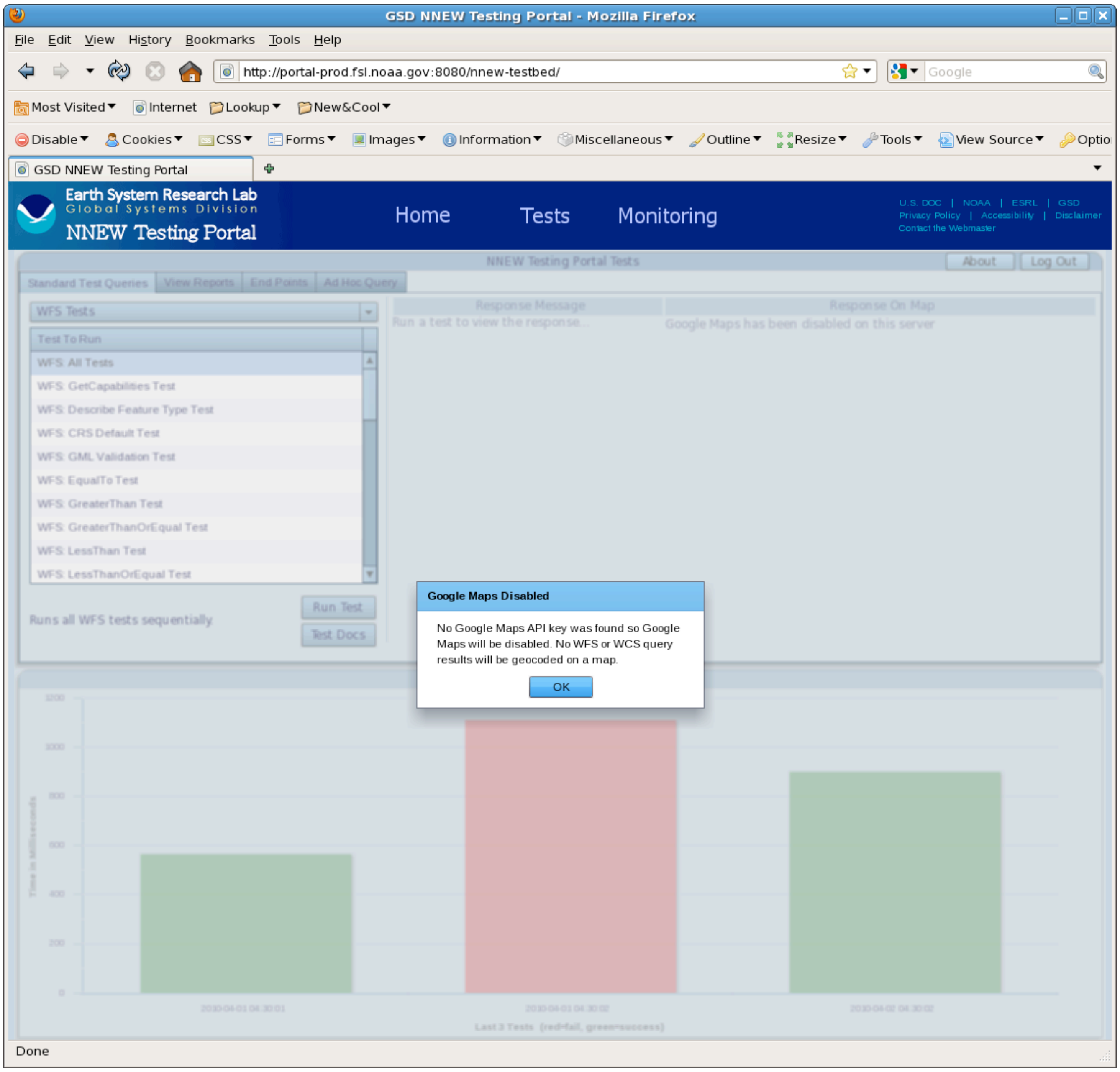
The performance monitoring page provides continuous, automated performance monitoring of the OGC web services.

More Information

You can learn more about NNEW from the NCAR public website.
NextGen is defined on the FAA's website.
GSD's NNEW TWiki (restricted access).

NNEW Testing Portal Home Page

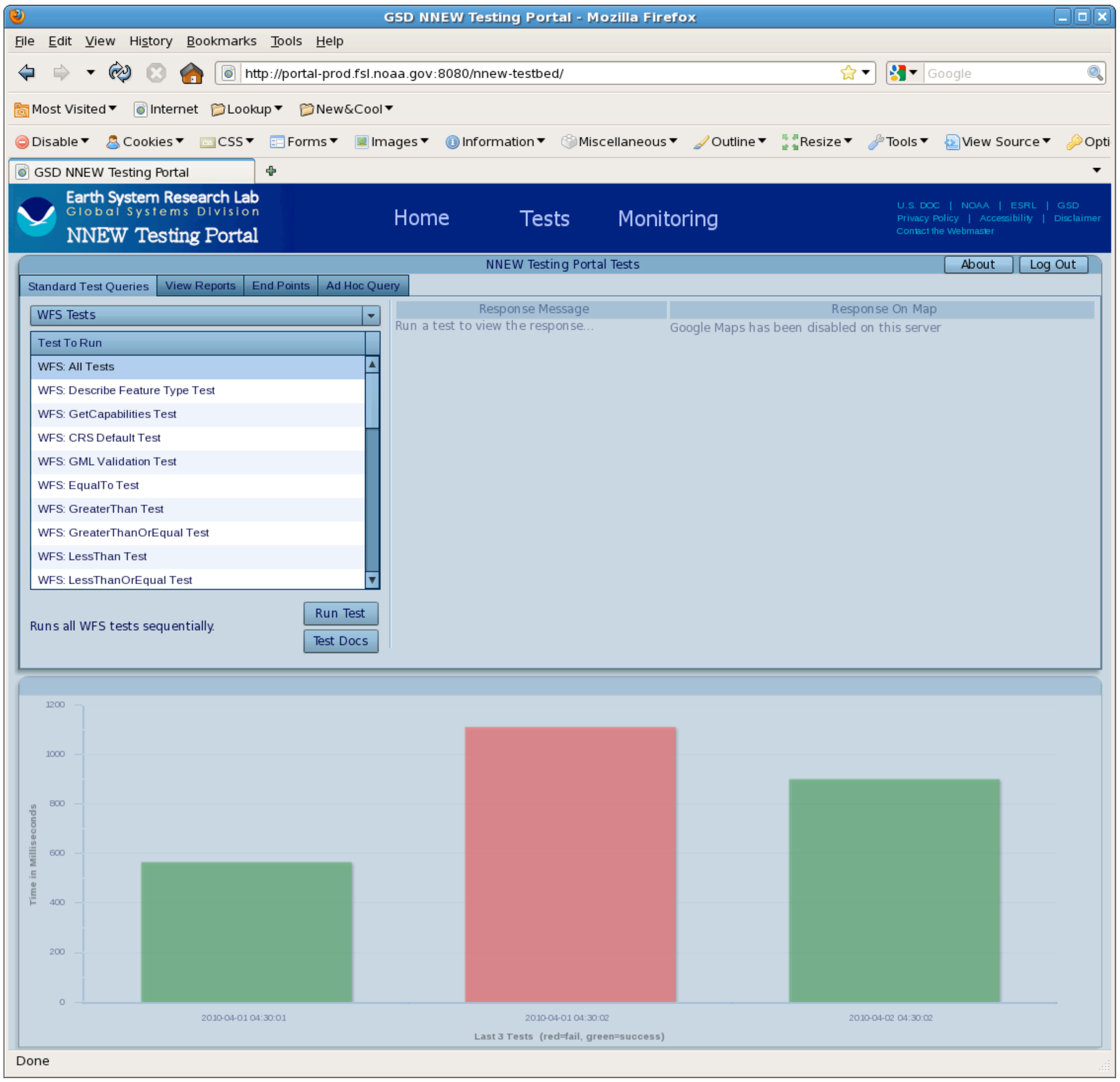
2. On the Testing Portal home page, click on 'Tests' in the title bar. A login window will be displayed
3. Enter user name and password. A window will be shown advising you that Google Maps has been disabled. This is normal when the Testing Portal is running at the Tech Center. Without access to Google Maps, the Testing Portal will not display the data returned from the WFSRI.


GSD NNEW Testing Portal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://portal-prod.fsl.noaa.gov:8080/nnew-testbed/
Earth System Research Lab
Global Systems Division
NNEW Testing Portal
Home
Tests
Monitoring
U.S. DOC | NOAA | ESRL | GSD
Privacy Policy | Accessibility | Disclaimer
Contact the Webmaster
NNEW Testing Portal Tests
About
Log Out
Standard Test Queries
View Reports
End Points
Ad Hoc Query
WFS Tests
Test To Run
WFS: All Tests
WFS: GetCapabilities Test
WFS: Describe Feature Type Test
WFS: CRS Default Test
WFS: GML Validation Test
WFS: EqualTo Test
WFS: GreaterThan Test
WFS: GreaterThanOrEqual Test
WFS: LessThan Test
WFS: LessThanOrEqual Test
Runs all WFS tests sequentially
Run Test
Test Docs
Response Message
Run a test to view the response...
Response On Map
Google Maps has been disabled on this server
Google Maps Disabled
OK
Time in Milliseconds
Last 3 Tests (red=fail, green=success)
Done

No Google Maps API key was found so Google Maps will be disabled. No WFS or WCS query results will be geocoded on a map.

NNEW Testing Portal: 'Google Maps Disabled' Notification

4. The Tests page will be displayed:


GSD NNEW Testing Portal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://portal-prod.fsl.noaa.gov:8080/nnew-testbed/
Google
Most Visited Internet Lookup New&Cool
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Opti
GSD NNEW Testing Portal
Earth System Research Lab
Global Systems Division
NNEW Testing Portal
Home
Tests
Monitoring
U.S. DOC | NOAA | ESRL | GSD
Privacy Policy | Accessibility | Disclaimer
Contact the Webmaster
NNEW Testing Portal Tests
About
Log Out
Standard Test Queries
View Reports
End Points
Ad Hoc Query
WFS Tests
Test To Run
Response Message
Response On Map
Run Test
Test Docs
1200
1000
800
600
400
200
0
Time in Milliseconds
2010-04-01 04:30:01
2010-04-01 04:30:02
2010-04-02 04:30:02
Last 3 Tests (red=fail, green=success)
Done

WFS: All Tests
WFS: Describe Feature Type Test
WFS: GetCapabilities Test
WFS: CRS Default Test
WFS: GML Validation Test
WFS: EqualTo Test
WFS: GreaterThan Test
WFS: GreaterThanOrEqual Test
WFS: LessThan Test
WFS: LessThanOrEqual Test

Run a test to view the response...

Google Maps has been disabled on this server

Runs all WFS tests sequentially.

NNEW Testing Portal Tests Page

There are four tabs on the upper left section of the page: Standard Test Queries, View Reports, End Points and Ad Hoc Query. The Standard Tests are used to test WFS functionality using a canned MDCR dataset, of known time and location. The View Reports page contains a list of reports generated by the Portal during testing, that the user can select and view. The End Points page displays the end point currently being accessed. The Ad Hoc Query page provides a list of xml queries that can be used as is or replaced by xml of the users choosing. The queries and the responses are both displayed. Each standard test has a corresponding ad hoc request. For the WFS Functional Tests, we'll be using a combination of Standard Tests and Ad Hoc Queries.

3.4.3.2 Test Procedure Example

This section contains an example test procedure. These instructions can be used for running any of the Portal's standard tests. After running each standard test, you can run the corresponding ad hoc test if you want to view the xml of the requests and responses.

Running the Standard Tests

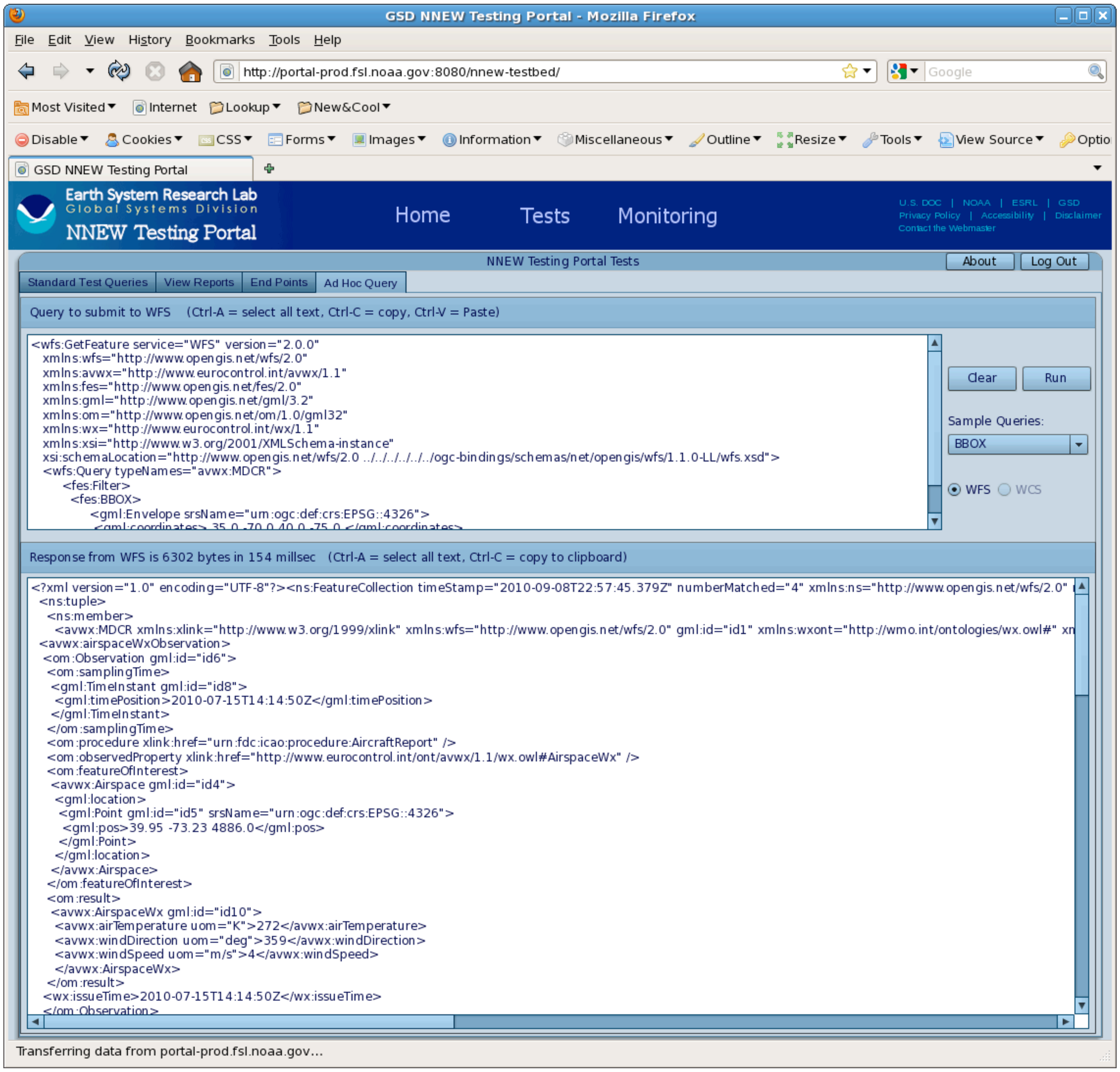
1. Click on the 'Standard Test Queries' tab.
2. Select one of the tests from the list, such as 'WFS: Bounding Box Test'. A description of the test will be displayed, including the requirement that is being tested.
3. Click on the 'Run Test' button at the bottom right of the Standard Test Queries box.

The Portal will run the test and report the results in the area to the right of the Standard Test Queries list. For example, the results for the WFS Bounding Box Test:

```
WFS: Bounding Box Test
        Status: SUCCESSFUL
Duration: 276 millisec
Geocodable: Yes
GF-bbox.xml query response from WFS contained the 4 expected features:
39.95 -73.23 4886.0
39.99 -73.22 4694.0
39.15 -74.87 10673.0
39.42 -74.62 10666.0
These features were all within the search BBOX ( 35.0,-70.0 40.0,-75.0 )
```

Running the Ad Hoc Tests

1. Click on the 'Ad Hoc Query' tab.
2. On the right side of the page under Sample Queries, click on Get Capabilities. A list of ad hoc queries will be displayed.
3. Select the test that corresponds to the Standard Test you just ran. For this example, the ad hoc query is 'BBOX'. The xml used for the Bounding Box query will be displayed.
4. Click on 'Run'. The xml response will be displayed:


NNEW Testing Portal Tests

Query to submit to WFS (Ctrl-A = select all text, Ctrl-C = copy, Ctrl-V = Paste)

```
<wfs:GetFeature service="WFS" version="2.0.0"
  xmlns:wfs="http://www.opengis.net/wfs/2.0"
  xmlns:avwx="http://www.eurocontrol.int/avwx/1.1"
  xmlns:fes="http://www.opengis.net/fes/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:om="http://www.opengis.net/om/1.0/gml32"
  xmlns:wx="http://www.eurocontrol.int/wx/1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs/2.0 ../../../../../../ogc-bindings/schemas/net/opengis/wfs/1.1.0-LL/wfs.xsd">
  <wfs:Query typeNames="avwx:MDCR">
      <fes:Filter>
        <fes:BBOX>
            <gml:Envelope srsName="urn:ogc:def:crs:EPSG::4326">
             <gml:coordinates> 35.0 -70.0 40.0 -75.0 </gml:coordinates>
```

Sample Queries: BBOX

[✓ WFS] WCS

Response from WFS is 6302 bytes in 154 millsec (Ctrl-A = select all text, Ctrl-C = copy to clipboard)

```
<?xml version="1.0" encoding="UTF-8"?><ns:FeatureCollection timeStamp="2010-09-08T22:57:45.379Z" numberMatched="4" xmlns:ns="http://www.opengis.net/wfs/2.0"
 <ns:tuple>
  <ns:member>
   <avwx:MDCR xmlns:xlink="http://www.w3.org/1999/xlink" xmlns:wfs="http://www.opengis.net/wfs/2.0" gml:id="id1" xmlns:wxont="http://wmo.int/ontologies/wx.owl#" xn
 <avwx:airspaceWxObservation>
  <om:Observation gml:id="id6">
   <om:samplingTime>
    <gml:TimeInstant gml:id="id8">
     <gml:timePosition>2010-07-15T14:14:50Z</gml:timePosition>
    </gml:TimeInstant>
   </om:samplingTime>
   <om:procedure xlink:href="urn:fdc:icao:procedure:AircraftReport" />
   <om:observedProperty xlink:href="http://www.eurocontrol.int/ont/avwx/1.1/wx.owl#AirspaceWx" />
   <om:featureOfInterest>
    <avwx:Airspace gml:id="id4">
     <gml:location>
      <gml:Point gml:id="id5" srsName="urn:ogc:def:crs:EPSG::4326">
       <gml:pos>39.95 -73.23 4886.0</gml:pos>
      </gml:Point>
     </gml:location>
    </avwx:Airspace>
   </om:featureOfInterest>
   <om:result>
    <avwx:AirspaceWx gml:id="id10">
     <avwx:airTemperature uom="K">272</avwx:airTemperature>
     <avwx:windDirection uom="deg">359</avwx:windDirection>
     <avwx:windSpeed uom="m/s">4</avwx:windSpeed>
    </avwx:AirspaceWx>
   </om:result>
  <wx:issueTime>2010-07-15T14:14:50Z</wx:issueTime>
  </om:Observation>
```

Transferring data from portal-prod.fsl.noaa.gov...

Results of the the BBOX Ad Hoc Query

3.4.4 Report Generation

The Testing Portal is capable of generating a report of the test results, based on the most recent execution of the WFS Functional Test Suite. For each test, the report contains the test name, a description, whether the test was successful, a description of the result, and the duration. To generate the report:

1. Go to the Standard Test Queries page.
2. Select WFS: All Tests
3. Click on 'Run Test' at the bottom left of the Standard Test Queries box. The Portal will run each of the tests in the suite, and display the results in the bar graph at the bottom of the page.

4. After the tests have finished, the Portal will display the test report.
5. Print out the report and add it to the test results documentation.

3.4.5 Test Procedures

This section contains descriptions of the WFSRI functional test procedures, and instructions for running them. For each procedure, the following information is provided:

- The procedure name
- A description of what the procedure does
- Requirements that are being tested by this procedure (listed in the GSD NNEW Test Matrix)
- Instructions for executing the test
- Expected results

Test Number: WFS-1

Test Name: GetCapabilities Test

Description

This test uses SOAP to test that the results from a GetCapabilities request conform to the WFS 1.1.0 schema. An XML Validator is used to validate the XML document against the WFS 1.1.0 schema.

Requirements

WFS 5.3.1.3 The WFSRI shall support Simple Object Access Protocol (SOAP) encoding over HTTP POST for the GetCapabilities operation.

WFS 5.3.2.1 The WFSRI shall provide an XML document indicating services, capabilities and features.

Instructions and Expected Results

#	Instructions	P	F
1	Click on the Standard Test Queries tab		
2	Select 'WFS: GetCapabilities Test' from the list of tests		
3	Click on 'Run Test'		
4	The following response message should be displayed to the right of the list of tests: WFS: GetCapabilities Test Status: SUCCESSFUL Duration: 446 millisec Geocodable: No getCapabilities.xml successfully validated against /usr/local/tomcat/ogc-bindings/trunk/schemas/net/opengis/wfs/2.0.0/wfs.xsd		

Comments:

Test Number: WFS-2

Test Name: CRS Default Test

Description

This test issues a WFS GetFeature request without specifying SRS/CRS. It verifies that data requested will be returned in the default SRS/CSR. The test procedure makes a GetFeature request and then searches for the srsName element, which contains the CRS name. The returned CRS name is compared with the default (*urn:ogc:def:crs:ESPG::4979).*

Requirements

WFS 5.1.3 The WFSRI shall return data using the default CRS if no CRS is specified.

Instructions and Expected Results

1. From the Standard Queries list, select 'CRS Default Test'

#	Instructions	P	F
1	From the Standard Queries list, select 'CRS Default Test' A description of the test should be displayed below the list of tests.		
2	Click on 'Run Test'		
3	A response message should be displayed to the left of the Tests to Run box. The message should look like this: *WFS: CRS Default Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: Yes* *Correct crsName in found response:* *Found:* *urn:ogc:def:crs:ESPG::4979*		

Comments:

Test Number: WFS-3

Test Name: Describe Feature Type Test

Description

This test uses SOAP to verify that the DescribeFeatureType request returns information about all available fields.

Requirements

WFS 5.3.3.3 The WFSRI shall support Simple Object Access Protocol (SOAP) encoding over HTTP POST for the DescribeFeature request.

Instructions and Expected Results

#	Instructions	Expected Results	P	F
1	Select 'WFS: Describe Feature Type Test' from the list of tests	A description of the test should be displayed below the 'Tests To Run' box.		
2	Click on 'Run Test'	The following response message should be displayed to the right of the 'Tests To Run' box: *WFS: Describe Feature Type Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: No* *describeFeatureType xml response from WFS did contain the following features: nawx:LightningFlashType*		

Comments:

Test Number: WFS-4
Test Name: GML Validation Test

Description
This test issues a WFS GetFeature request for TAF data, and validates the response against gml 3.1.1 (OGC 03-105R1).

Requirements
WFS 4.3.1 The WFSRI shall require the use of GML(OGC 03-105R1) to express features within the interface and at a minimum, be used to present features. The GML version will conform to the version specified in WFS specification.

Instructions and Expected Results

#	Instructions	P	F
1	Click on the Standard Test Queries tab		
2	Select WFS: GML Validation Test		
3	Click on 'Run Test'		
4	Verify that the following response message is displayed: WFS: GML Validation Test Status: SUCCESSFUL Duration: n millisec Geocodable: No GF-GML-validate.xml successfully validated against [/usr/local/tomcat/ogcbindings/trunk/schemas/net/opengis/gml/3.1.1/base/gml.xsd, /usr/local/tomcat/ogc-bindings/trunk/schemas/net/opengis/wfs/2.0.0/wfs.xsd, /usr/local/tomcat/ogc-bindings/trunk/schemas/int/eurocontrol/avwx/1.1.1/avwx.xsd]		

Comments:

Test Number: WFS-5
Test Name: Equal To Test

Description
This test WFS verifies that a PropertyIsEqualTo filter returns correct features. A GetFeature request of MDCRS features that match a <fes:PropertyIsEqualTo> filter is issued.

Requirements
5.0.1 The WFSRI shall support the WFS version 1.1.0/2.0 specifications.

Instructions and Expected Results

#	Instructions	P	F
1	Click on the Standard Test Queries tab		
2	Select 'WFS: Equal To Test' from the list of tests		
3	Click on 'Run Test'		
4	Verify that the following response message is displayed: *WFS: EqualTo Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: Yes* *GF-EqualTo.xml WFS query successfully returned 11 features*		

Comments:

Test Number: WFS-6
Test Name: Greater Than Test

Description
This test verifies that a PropertyIsGreaterThan filter returns correct features. A GetFeature request of MDCR features that match a <fes:PropertyIsGreaterThan> filter is issued.

Requirements
5.0.1 The WFSRI shall support the WFS version 1.1.0/2.0 specifications.

Instructions and Expected Results

#	**Instructions**	**P**	**F**
1	Select 'WFS: Greater Than Test' from the Standard Test Queries list		
2	Click on 'Run Test'		
3	Verify that the following response message is displayed: *WFS: Greater Than Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: Yes* *GF-greaterThan.xml WFS query successfully returned 47 features*		

Comments:

Test Number: WFS-7
Test Name: Greater Than Or Equal Test

Description
This test verifies that a PropertyIsGreaterThanOrEqualTo filter returns the correct features. A GetFeature request of MDCR features that match a <fes:PropertyIsGreaterThanOrEqualTo> filter is issued.

Requirements
5.0.1 The WFSRI shall support the WFS version 1.1.0/2.0 specifications.

Instructions and Expected Results

#	Instructions	P	F
1	Select 'WFS: GreaterThanOrEqual Test' from the Standard Test Queries list		
2	Click on 'Run Test'		
3	Verify that the following response message is displayed: *WFS: GreaterThanOrEqual Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: Yes* *GF-greaterThanOrEqual.xml WFS query successfully returned 183 features*		

Comments:

Test Number: WFS-8
Test Name: Less Than Test

Description
This test verifies that a PropertyIsLessThan filter returns correct features. A GetFeature request of MDCR features that match a <fes:PropertyIsLessThan> filter is issued.

Requirements
5.0.1 The WFSRI shall support the WFS version 1.1.0/2.0 specifications.

Instructions and Expected Results

#	Instructions	P	F
1	Select 'WFS: LessThan Test' from the list of tests		
2	Click on 'Run Test'		
3	Verify that the following response message is displayed: *WFS: LessThan Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: Yes* *GF-LessThan.xml WFS query successfully returned 1 features*		

Comments:

Test Number: WFS-9
Test Name: LessThanOrEqual Test

Description
This test verifies that a PropertyIsLessThanOrEqualTo filter returns correct features. A GetFeature request of MDCR features that match a <fes:PropertyIsLessThanOrEqualTo> is issued.

Requirements
5.0.1 The WFSRI shall support the WFS version 1.1.0/2.0 specifications.

Instructions and Expected Results

#	Instructions	P	F
1	Select 'WFS: LessThanOrEqual Test' from the Standard Test Queries list		
2	Click on 'Run Test'		
3	Verify that the following response message is displayed: *WFS: LessThanOrEqual Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: Yes* *GF-LessThanOrEqual.xml WFS query successfully returned 1 features*		

Comments:

Test Number: WFS-10
Test Name: Bounding Box Test

Description
This test verifies that for a given lat-lon box, expected elevations are returned. A GetFeature request for MDCR features within a BBOX (35.0,-70.0 40.0,-75.0) is issued. The features returned are compared to the known records within that bounding box.

Requirements
5.0.1 The WFSRI shall support the WFS version 1.1.0/2.0 specifications.

Instructions and Expected Results

#	Instructions	P	F
1	Select 'WFS: Bounding Box Test' from the list of tests		
2	Click on 'Run Test'		
3	Verify that the following response is displayed: *WFS: Bounding Box Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: Yes* *MDCRS_bbox.xml query response from WFS contained the 4 expected features:* *39.95 -73.23 4886.0* *39.99 -73.22 4694.0* *39.15 -74.87 10673.0* *39.42 -74.62 10666.0* These features were all within the search BBOX (35.0,-70.0 40.0,-75.0)		

Comments:

Test Number: WFS-11
Test Name: CONUS Box Test

Description
WFS GetFeature test verifies that for CONUS lat-lon box, expected number of features are returned. A GetFeature request of MDCR features is issued, for within the CONUS BBOX (20.3,-68.8 47.7,-136.1) that have timePosition "2010-07-15T14:07:10Z". The features returned are compared to the known records within that bounding box for that timePosition.

Requirements
5.0.1 The WFSRI shall support the WFS version 1.1.0/2.0 specifications.

Instructions and Expected Results

#	Instructions	P	F
1	Click on the Standard Test Queries tab		
2	Select 'WFS: CONUS Box Test'		
3	Click on 'Run Test'		
4	Verify that the following response message is displayed: *WFS: MDCRS CONUS Box Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: Yes* *GF-CONUS-bbox.xml query response from WFS contained the 9 expected features:* *20.78 -73.8 9449.0* *40.54 -73.68 1448.0* *26.02 -80.28 3139.0* *26.16 -80.46 5197.0* *33.73 -81.53 3932.0* *37.98 -84.83 6715.0* *32.7 -97.24 3173.0* *32.71 -97.57 4380.0* *47.54 -121.89 3932.0* *These features were all within the search BBOX (20.3,-68.8 47.7,-136.1)*		

Comments:

Test Number: WFS-12
Test Name: Point Test

Description
This test issues a WFS GetFeature request for air temperature for a specified point (X, Y, Z) and verifies that it is returned successfully. Only one feature should be returned.

Requirements
WFS 4.2.1.1 The WFSRI shall support geometric filtering of features, including the following: Point. A feature may be subsetted by requesting a single point specified by X, Y, and Z values.
WFS 5.0.1 The WFSRI shall support the WFS version 1.1.0/2.0 specifications.

Instructions and Expected Results

#	Instructions	P	F
1	Select 'WFS: Point Test' from the Standard Test Queries list		
2	Click on 'Run Test'		
3	Verify that the following response message is displayed: WFS: Point Test Status: SUCCESSFUL Duration: n millisec Geocodable: Yes GF-point.xml WFS query successfully returned 1 features		

Comments:

Test Number: WFS-13
Test Name: Not Equal To

Description
This test verifies that PropertyIsNotEqualTo filter returns correct features.

Requirements
WFS 5.0.1 The WFSRI shall support the WFS version 1.1.0/2.0 specifications.

Instructions and Expected Results

#	Instructions	P	F
1	Select 'WFS: Not Equal To' from the Standard Test Queries list.		
2	Click on 'Run Test'		
3	Verify that the following response message is displayed: WFS: NotEqualTo Test Status: SUCCESSFUL Duration: n millisec Geocodable: Yes GF-NotEqualTo.xml WFS query successfully returned 182 features		

Comments:

Test Number: WFS-14
Test Name: Subset By Zero Test

Description
This test issues a WFS GetFeature test without specifying properties in the query filter. The test verifies that all properties are returned. A DescribeFeatureType request is issued for esrl:mdcr data and the number of fields is ascertained. This value is compared to the expected number to determine whether the test succeeded.

Requirements
WFS 4.2.1 The WFSRI shall allow subsetting by zero or more fields (e.g., temperature, reflectivity). If no fields are indicated, all available fields shall be returned. All fields specified as mandatory by the data provider will be returned, irrespective of the user request. The mandatory fields will be specified by the data provider at the time of registration of the feature type.

Instructions and Expected Results

#	Instructions	P	F
1	Select 'WFS: Subset By Zero Test' from the Standard Test Queries list		
2	Click on 'Run Test'		
3	Verify that the following response message is displayed: *WFS: Subset By Zero Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: Yes* *GF-GetFeatureSubsetByZeroFields.xml response from WFS contained all of the following features:* *avwx:rawText avwx:aerodromeWxForecast avwx:appliesTo* *avwx:issueTime* *avwx:issueTime* *avwx:stationId* *avwx:type*		

Comments:

Test Number: WFS-15
Test Name: Temporal Subset List Times Test

Description
This test requests a list of times and verifies that the WFSRI returns the correct time.

Requirements

WFS 4.2.2.1 The WFSRI shall be capable of providing a list of valid times from datasets that are temporally aggregated.

Instructions and Expected Results

#	Instructions	P	F
1	Select 'WFS: Temporal Subset List Times Test" from the Standard Test Queries list.		
2	Click on 'Run Test'		
3	Verify that the following response message is displayed: WFS: Temporal Subset List Times Test Status: SUCCESSFUL Duration: n millisec Geocodable: Yes GF-SubsetListTimes.xml WFS query successfully returned 64 features		

Comments:

Test Number: WFS-16
Test Name: Temporal Subset Multiple Time Test

Description
This test issues a WFS GetFeature request for multiple times and verifies that multiple matching times are returned. The test validates the results against two schemas, then verifies that the results only have expected times in them and that the times are the same as the original query (thus verifying it returned the correct data).

Requirements
WFS 4.2.2.3 The WFSRI shall provide the capability to constrain features by requesting a range of valid times.

Instructions and Results

#	Instructions	Expected Results	P	F
1	Select 'WFS: Temporal Subset Multiple Time Test' from the list of tests	A description of the test should be displayed below the 'Tests To Run' box.		
2	Click on 'Run Test'	The following response message should be displayed to the right of the 'Tests To Run' box: *WFS: Temporal Subset Multiple Time Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: No* *GetFeatureTemporalSubsettingMultiTime.xml successful. All time(s) in response match the expected time: 2010-05-26T12:45:00Z*		

Comments:

Test Number: WFS-17
Test Name: Temporal Subset Single Time Test

Description
This test issues a WFS GetFeature request for a single time value (2010-05-26T12:45:00Z) and verifies that only data from that single time are returned.

Requirements
WFS 4.2.2.2 The WFSRI shall provide the capability to constrain features by requesting a single valid time.

Instructions and Expected Results

#	Instructions	P	F
1	Select 'WFS: Temporal Subset Single Time Test' from the list of tests A description of the test should be displayed below the 'Tests To Run' box.		
2	Click on 'Run Test'		
3	The following response message should be displayed: *WFS: Temporal Subset Single Time Test* *Status: SUCCESSFUL* *Duration: n milisec* *Geocodable: No* *GetFeatureTemporalSubsettingSingleTime xml successful.* *All time(s) in response match the expected time: 2010-05-26T12:45:00Z*		
4	Click on the Ad Hoc Query tab		
5	Select 'Subset One Time' from the Sample Queries list		
6	View the xml query displayed in the upper window		
7	Click on Run		
8	Verify that the xml response contains only one issue time: <avwx:issueTime>2010-05-26T12:45:00Z</avwx:issueTime>		

Comments:

Test Number: WFS-18
Test Name: TAF Greater Than Test

Description
A WFS GetFeature test verifies that a TAF PropertyIsGreaterThan Filter returns the correct features. The GetFeature request selects TAF records with an issue time greater than a constant (time).

Requirements
5.0.1 The WFSRI shall support the WFS version 1.1.0/2.0 specifications.

Instructions and Expected Results

#	Instructions	P	F
1	Click on the Standard Test Queries tab		
2	Select 'WFS: TAF Greater Than Test'		
3	Click on 'Run Test'		
4	Verify that the following response message is displayed: *WFS: TAF GreaterThan Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: Yes* *TAFgreaterThan.xml WFS query successfully returned 183 features*		

Comments:

Test Number: WFS-19
Test Name: Vertical 2D Test

Description
This test verifies that all TAF records within a given GML exterior/linear ring are returned.

Requirements
WFS 4.2.1.1 The WFSRI shall support geometric filtering of features, including the following: Vertical 2-D Cross-section. For this case, a vertical slice is taken through a feature. The path for the cross-section is defined by a set of XY waypoints and a sample density.

Instructions and Expected Results

#	Instructions	P	F
1	Select 'WFS: Vertical 2D Test' from the Standard Test Queries list.		
2	Click on 'Run Test'		
3	Verify that the following response message is displayed: WFS: Vertical 2D Test Status: SUCCESSFUL Duration: n millisec Geocodable: Yes GF-Vertical2D.xml WFS query successfully returned 24 TAF records within given GML exterior/linear ring		

Comments:

Test Number: WFS-20
Test Name: Volume 3D Test

Description
This test verifies filtering by GML composite surface (3D bounding box), returning TAF records within the given volume.

Requirements
WFS 4.2.1.1 The WFSRI shall support geometric filtering of features, including the following: 3D Volume. For this case, a 3-dimensional bounding box may be specified with dimensions defined by X, Y, Z value (such as latitude, longitude and altitude) ranges. A user can query for data that coincides with this volume

Instructions and Expected Results

#	Instructions	P	F
1	Select 'WFS: Volume 3D Test' from the Standard Test Queries list.		
2	Click on 'Run Test'		
3	Verify that the following response message is displayed: WFS: Volume 3D Test Status: SUCCESSFUL Duration: n millisec Geocodable: Yes GF-Volume3D.xml WFS query successfully returned 24 TAF records within given GML composite surface (box)		

Comments:

Test Number: WFS-21
Test Name: WXXM Validation Test

Description
WFS GetFeature request is validated against WFS and AVWX schemas.

Requirements
WFS 4.1.4 The WFSRI shall provide output in an ISO 19139 format.

Instructions and Expected Results

#	Instructions	P	F
1	Select 'WFS: WXXM Validation Test' from the list of tests		
2	Click on 'Run Test'		
3	Verify that the following response message is displayed: *WFS: WXXM Validation Test* *Status: SUCCESSFUL* *Duration: n millisec* *Geocodable: No* *GetFeatureTemporalSubsettingSingleTime.xml successfully validated against [/usr/local/tomcat/ogc-bindings/trunk/schemas/net/opengis/wfs/2.0.0/wfs.xsd, /usr/local/tomcat/ogc-bindings/trunk/schemas/net/opengis/wfs/2.0.0/wsdl/wfs-util.xsd, /usr/local/tomcat/ogc-bindings/trunk/schemas/int/eurocontrol/avwx/1.1.1/avwx.xsd]*		

Comments:

4 WCSRI 2.0 Functional and Implementation Verification Test Procedures

4.1 Overview

This document describes the test plan NCAR deliverables for the 2010 Capability Evaluation. NCAR's key deliverables include the gridded and non-grided datasets and Web Coverage Services Reference Implementation (WCSRI) Version 2.0.

4.2 WCSRI Overview

The WCSRI provides services to access gridded data for the NNEW 4-D Weather data cube. WCSRI gives data providers the ability to configure coverages, data directories, metadata and coverage specific information. In addition, it provides data retrieval services and coverage subsetting capabilities by field, time and geographical constraints.

For fiscal year 2010, WCSRI has focused on distribution of services by adding the ability to host origin and distribution servers. This allows data to be distributed over various hosts and a server to direct the data access request to upstream servers. WCSRI has also added support for publish/subscribe way of communication allowing clients to talk to the servers asynchronously, in addition to the request/reply model.

4.3 WCSRI System Requirements

System dependencies are required to run the demonstration applications. These include computer memory, network connectivity, libraries and utility packages. All of these items must be installed prior to WCSRI 2.0 installation.

4.3.1 Hardware Requirements

The test system will ideally consist of two machines. One machine should be 64-bit architecture running some variation of the LINUX Operating System. This machine will be known as the "WCS Server" in the remainder of this document, and will be used for running the WCSRI server instance. The other machine, known as the "Demo Machine", will be used to run the validation suite of tests, and must also be Linux due to complications with running SoapUI on a Windows machine. We recommend at least 2GB of RAM for each machine.

4.3.2 Software Requirements

The software requirements for the test procedures are as follows:

Software Package	Description
wcsri2.1 (latest version available)	Web Coverage Service Reference Implementation. This is the primary software being tested and is available at: https://wiki.ucar.edu/display/NNEWD/Reference+Implementations It is recommended that the validation suite be used to ensure a functioning installation.
soapUI 3.5 or greater	OpenSource software that provides a UI to test WSDL http://sourceforge.net/projects/soapui/
Java 6.0 Runtime Environment	Java runtime, required for all test procedures http://java.sun.com/
FUSE Enterprise Service Bus (version 4.2.0-fuse-01-00)	Required for the WCSRI. https://wiki.ucar.edu/display/NNEWD/Reference+Implementations
NetCDF 4 C library	Required for the WCSRI. Installation instructions included in doc/INSTALLATION.txt in the WCSRI release file.
HDF 5 C library	Required for the WCSRI. Installation instructions are included in doc/INSTALLATION.txt in the WCSRI release.
NetCDF 4 Java Native Interface/C bindings	Required for the WCSRI. https://wiki.ucar.edu/display/NNEWD/Reference+Implementations
Mozilla Firefox (or other browser that can render XML)	Internet browser for running simple WCS requests. http://www.mozilla.com/
ToolsUI	Interactive thick-client Java Web Start display for visualizing NetCDF data. http://www.unidata.ucar.edu/software/netcdf-java/

The test procedures assume WCSRI has been successfully installed on a system and validation (via the validation package) has been run on the test system. The WCSRI Version 2 release is available in two bundles:

wcsri-<version>-release.tar.gz	Consists of WCSRI server side software
wcsri-<version>-validation.tar.gz	Consists of validation configuration, datasets and client-side test scripts.

4.4 WCSRI 2.0 Test Procedures

Our goal for FY10 tests has been to get as much requirements coverage as possible and provide automated test suites. Test cases in this document have been organized as below.

Test Case 1: Validate WCSRI Server as the Origin Server via SOAP UI

This set of test cases will setup WCSRI server as an origin server for the test dataset. We will run the Test Suite in SOAPUI to validate data retrieval functions including the new operation “GetMetadata”. We will also verify various configurable properties for the datasets.

Test Case 2: Validate Key/Value based HTTP requests

This test case will verify that the WSDL and key/value pair HTTP requests work as expected.

Test Case 3: Validate Distribution WCSRI Server as a Repeater

In this set of test cases, we will setup the WCSRI Server as a distributor for an origin server at NCAR. In addition, the test case will also validate the repeater functionality and the server’s ability to cache data locally.

Test Case 4: Validate Distribution WCSRI Server as a Delegator

This test will be a slight variation on Test Case 4, where the server will be setup as a delegator to an Origin Server running at NCAR. In this case, we will validate all client requests for the test dataset are delegated to the origin server.

Any questions or problems encountered with the tests or test procedures should be reported to wcsri-support@rap.ucar.edu.

Definitions

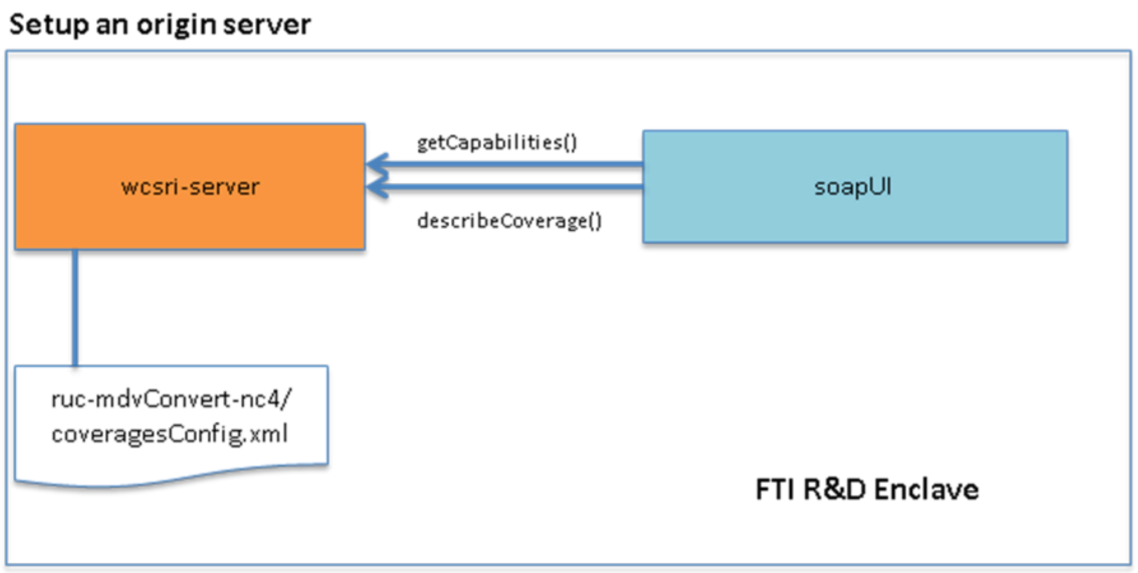
- $WCS_VALIDATION – an environment variable on the WCS server machine and the Demo machine, that defines the directory location for the unbundled WCS validation file (ie: wcsri-2.1-validation.tar.gz). On the WJHTC WCS server machine (ie: “wcs”) this directory is: `/home/users/wcsri/capEvalFY10/wcsriValidation`. On the Demo machine, this directory is: `/home/users/nwec/capEvalFY10/wcsriValidation.`
- `$FUSE_ESB_HOME` – an environment variable on the WCS server machine, that defines the directory location for the Fuse/Servicemix installation.
- $SOAPUI_HOME - an environment variable on the Demo machine, that defines the directory location for the SoapUI installation.
- $NC4_HOME - an environment variable on the Demo machine, that defines the directory location for the Netcdf executables (eg: ncdump) installation.

4.4.1 Test Case 1: Validate WCSRI Server by sending SOAP requests

Purpose and Scope:
Validate WCSRI Server using SoapUI. Show various configurable parameters and their verification. Demonstrate the new operation called "`GetMetadata`".

Reference Documents:
WCSRI Installation Document

Test Description:
In this test, we will use a SoapUI client on the Demo machine to test the WCSRI installed on the WCS server machine. The tests will verify Soap request/reply functionality against the test dataset distributed with the server installation. The diagram below shows the setup for this test case. In addition, we will verify the GetMetadata operation and review various configurable parameters as they relate to WCSRI V1 and V2 requirements.



System Under Test:
WCSRI Version 2.1

Test Setup:

- WCSRI Validation Suite version 2.1
    - The contents of the wcsri-2.1-validation.tar.gz should be extracted on both the WCS server machine and the Demo machine. The directory that it's extracted to will be referred to as $WCS_VALIDATION below.
- On the WCS Server machine, WCSRI version 2.1 must be installed and the following steps performed:

- Start this test procedure with a clean servicemix deployment, by performing the following steps:

  cd $FUSE_ESB_HOME
  rm –rf data/ deploy/ pubsub_data/

- Start this test procedure with a clean test harness by removing any files called "subscription.properties" beneath the $WCS_VALIDATION and $NEW_DATA_DIR directories. This can be done with the following commands (from the WCS Server):

```
cd $WCS_VALIDATION
find . –name subscription.properties | xargs rm
cd $NEW_DATA_DIR;
find . –name subscription.properties | xargs rm
```

- On the Demo machine, SoapUI 3.5.1 must be installed and the following steps performed:

  - Verify that the following line has been *uncommented* in $SOAPUI_HOME/bin/soapui.sh:

```
JAVA_OPTS="$JAVA_OPTS -Dsoapui.jxbrowser.disable=true"
```

  - Verify that validation-2.1-jar-with-dependencies.jar and wx-consumer-wcs-all.jar have been copied to the $SOAPUI_HOME/bin/ext directory.

    ls $SOAPUI_HOME/bin/ext

  - If files do not exist, perform the following steps:

```
cp $WCS_VALIDATION/soapUI/validation/target/validation-2.1-jar-with-
dependencies.jar $SOAPUI_HOME/bin/ext

cp $WCS_VALIDATION/wx-consumer-wcs/wx-consumer-wcs-all.jar
$SOAPUI_HOME/bin/ext
```

Test Equipment:

- WCS Server - the Linux server called "wcs" at the WJHTC for running the WCSRI instance.
- The "Demo" machine – Linux machine for running the SoapUI client.

Personnel:
Tests conducted by test engineers at FAA Tech Center. NCAR support provided by wcsri-support@rap.ucar.edu

Location:
FAA Tech Center

Test Conduct:

Safety Considerations:
See error log files available at /tmp/wcsri.log

Requirements under test (directly or indirectly):

Number	Brief Description
4.1.1	Provide ability for the Service Provider to specify the available coverages that will be served by the WCSRI.
4.1.2	Provide ability to configure id, name, directory and metadata for each coverage.
4.1.5	Provide lifecycle management, namely scrubbing.
4.2.1	Provide subsetting capability by field.
4.2.1.1	Provide geometric subsetting of coverage volumes.
4.2.3.1	Provide a list of valid times representing data availability.
4.2.3.2	Provide temporal subsetting by a single valid time.
4.2.3.3	Constrain a coverage by an analysis time and valid time combination (e.g., retrieve a coverage subset that was generated as part of the 12:00Z model run cycle, and is valid for 14:00Z)
4.2.4.1	Provide the ability to return time series of coverages.
4.2.5.1	Support temporal trajectories.
4.3.1	Support CF-Netcdf4 as a native file format.
4.3.2	Expose CF-Netcdf4 at the protocol level.
4.3.3	Support GRB2 as a native file format.
5.2.1	Provide means to enable/disable operations by encoding type.
5.3.2	Support for SOAP GetCapabilities.
5.4.1.1	Statically configure metadata describing GetCapabilities elements.
5.4.2.1	Provide a means for Service provider to statically configure endpoints.
5.4.3.1	Ability to configure path for storing temporary files and scrubbing information.
5.4.4.1	Provide ability to get coverage identifier, bounding box, supported CRS and projections and native CRS.
5.4.4.2	Support “application/x-NetCDF” as a supportedFormat.
5.5.1	Support for SOAP DescribeCoverage.
5.6.2.1	Provide ability to configure metadata for each field.
5.6.2.2	Support “nearest neighbor” interpolation.
5.6.2.3	Provide Units of Measure for fields within DescribeCoverage response.
5.7.1	Support for SOAP GetCoverage.
5.9.1	Support for ISO-19139 GetMetadata operation.
6.1	Description of procedure for adding and removing coverages.
6.2.7	Support for configuring architectural patterns.
10.1	WCSRI should be capable of self-test diagnostic following installation.

Test Procedures:

Configure WCSRI Server:

( ) 1. Use ssh to connect to the WCS server, that has an installed and validated WCSRI server. For example:

```
ssh wcsri@wcs
```

( ) 2. Edit $FUSE_ESB_HOME/etc/wcsri.cfg. For example:

```
vi $FUSE_ESB_HOME/etc/wcsri.cfg
```

( ) 3. Set the `wcs.rootDataDir` property to the value of your $WCS_VALIDATION/testData directory. For example:

```
wcs.rootDataDir=/home/users/wcsri/capEvalFY10/wcsriValidation/testData
```

Also note the value of the the `wcs.frontend.publish.url` (____________________) and `wcs.frontend.external.url` (__________________________________) properties since they will be used in future test steps.

( ) 4. Since the default values are fine for many of the properties in the wcsri.cfg file, it will suffice to simply review the ones listed below in the table. Each of the properties mentioned below is pertinent to meeting a specific WCSRI requirement, as listed as the "Purpose" for the property.

Property	**Purpose**
`wcs.tempDir`	Specifies the directory for temporary files (for Req. 5.4.3.1)
`wcs.servicesMetaFile`	Specifies the path for the servicesMeta.xml file (for Req 5.4.1.1)
`wcs.kvp.disabled`	Enables/disables KVP Encoding Type (Req 5.2.1)
`wcs.securityEnabled`	Enables/disables security (eg: authentication)
`wcs.frontend.publish.url`	Configure Endpoints (for Req. 5.4.2.1)

( ) 5. Edit $WCS_VALIDATION/testData/servicesMeta.xml. Review the file for the following elements:
( ) `ServiceIdentification`
( )`ServiceProvider`
( ) `OperationsMetaData`

The purpose of this file is to allow the Service Provider to statically enter metadata about their WCS server (Req 5.4.1.1). Information entered here will be presented as XML within a `GetCapabilities` response.

Optional: Edit the `<ServiceProvider>` element's `<ProviderName>` element and set it to something pertinent to your organization. For example:

```
<ows:ProviderName>WJHTC</ows:ProviderName>
```

( ) 6. View $WCS_VALIDATION/testData/ruc-mdvConvert-nc4/coverageMeta1.xml. The purpose of this file is to allow the Service Provider to statically enter metadata about this particular coverage offered by their WCS server (Req 4.1.2). Information entered here will be presented as XML within a DescribeCoverage response.

Review the `CoverageDescription` element (Req 4.1.2) for the following elements:

( ) `Title`
( ) `Abstract`
( )`Keywords`
( ) `Metadata`

These elements allow the Service Provider to configure such static information. The remainder of the `CoverageDescription` element (eg: Domain, Range) is dynamically generated by the server and should not appear in the static coverageMeta1.xml file.

Optional: Edit the <Title> element and change it to something relevant, such as:

```
<ows:Title>WJHTC Test Dataset - RUC</ows:Title>
```

( ) 7. View $WCS_VALIDATION/testData/ruc-mdvConvert-nc4/coveragesConfig.xml. Each coveragesConfig.xml specifies the root of a data source (eg: set of data files) to be used by the server (Req 4.1.1 and 4.1.2). The server looks for these files each of which contains configuration information for one or more coverages, whose data lies below the directory where the configuration file is found.

Review the following settings:

	Data Blocks	**Purpose**
( )	`DataSource -> type and fileExtension` attributes	Type FileSystem (Req 6.2.7) and native data file extension (Req 4.3.1 and 4.3.3)
( )	`ValidTimesType` element	DescribeCoverage responses will either contain a list of valid times, valid/gen time combinations, or a time range describing the availability of data (Req 4.2.3.1)
( )	`Janitor` element	Data cleanup (Req 4.1.5 and 5.4.3.1)
( )	`Coverage -> Id` element	Identifier for the coverage(s) (Req 4.1.2)
( )	`Coverage -> Field`	`Field` specification (Req 4.2.1)

	element	
( )	`Coverage -> Field's` child elements (eg: `Abstract`)	`Field` Metadata (Req 5.6.2.1)

( ) 8. Edit $WCS_VALIDATION/testData/RUC20P/coveragesConfig.xml. Verify that the `OriginServerWcsEndpoint` points to the value of the `wcs.frontend.publish.url` property found in the $FUSE_ESB_HOME/etc/wcsri.cfg file. For example:

```
<OriginServerWcsEndpoint>http://0.0.0.0:8280/wcs/soap</OriginServerWcsEndpoin
t>
```

The `OriginServerWcsEndpoint` element is used for configuring “Repeater” and “Delegator” datasets and is required for the validation tests to succeed, though it will not be described until the latter Test Cases.

( ) 9. Edit $WCS_VALIDATION/testData/Delegator/coveragesConfig.xml. Verify that the `OriginServerWcsEndpoint` points to the value of the `wcs.frontend.publish.url` property found in the $FUSE_ESB_HOME/etc/wcsri.cfg file. For example:

```
<OriginServerWcsEndpoint>http://localhost:8280/wcs/soap</OriginServerWcsEndpo
int>
```

The `OriginServerWcsEndpoint` element is used for configuring “Repeater” and “Delegator” datasets and is required for the validation tests to succeed, though it will not be described until the latter Test Cases.

( ) 10. Verify that servicemix is not running, and if it is, shut it down or kill it:

```
ps -ef | grep servicemix
kill -9 [the process number of servicemix]
```

Verify that “data”, “deploy” and “pubsub_data” directories (and their contents) have been removed from $FUSE_ESB_HOME so that we start these test procedures with a clean instance of servicemix.

Remove any old log files as below:

```
rm /tmp/wcsri.log
```

( ) 11. Start servicemix by issuing the following command:

```
cd $FUSE_ESB_HOME
./bin/servicemix
```

Wait for approximately 30 seconds.

( ) 12. At the servicemix prompt, type:

```
features:install wcsri
```

Wait for approximately 2 minutes. Servicemix should now be successfully running with an installed instance of the WCSRI.

SoapUI Setup on the Demo Machine:

( ) 13. Verify that validation-2.1-jar-with-dependencies.jar and wx-consumer-wcs-all.jar have been copied to the $SOAPUI_HOME/bin/ext directory.

ls $SOAPUI_HOME/bin/ext

- If files do not exist, perform the following steps:

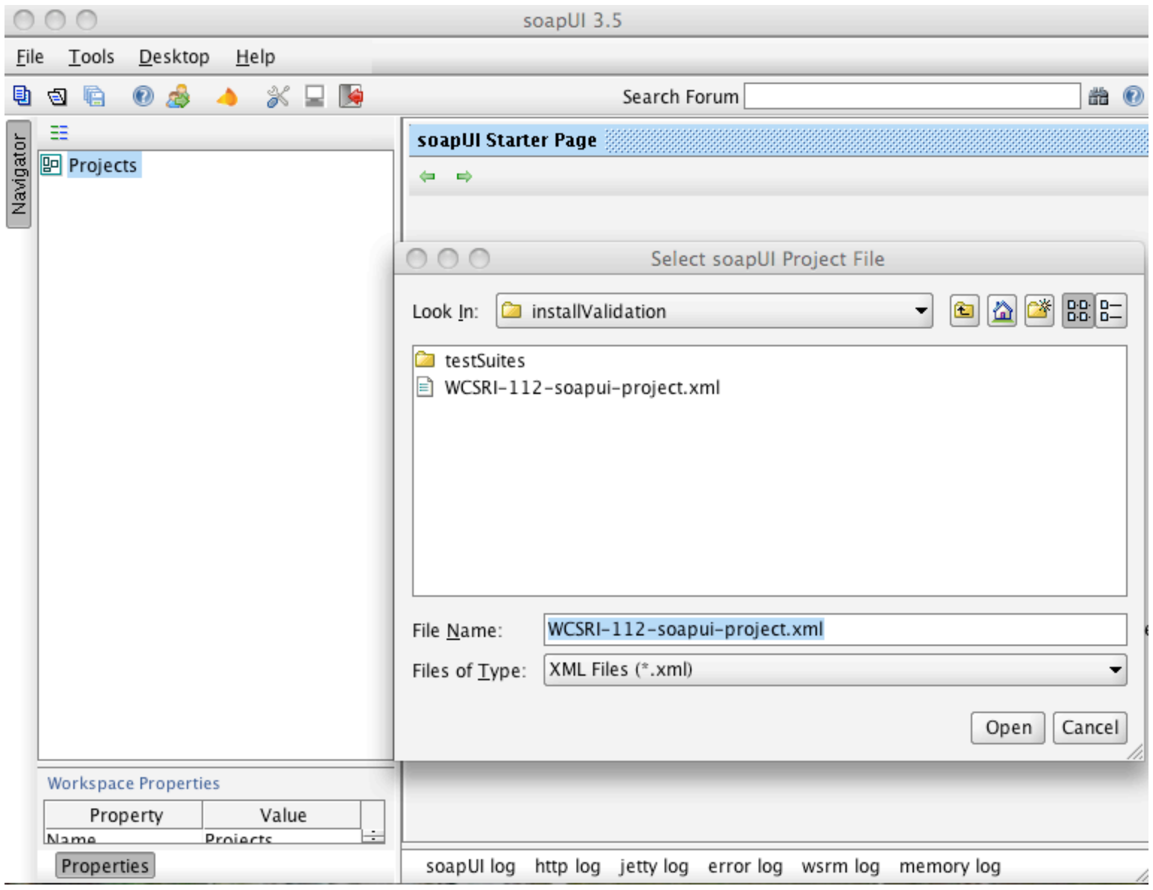
```
cp $WCS_VALIDATION/soapUI/validation/target/validation-2.1-jar-with-
dependencies.jar $SOAPUI_HOME/bin/ext

cp $WCS_VALIDATION/wx-consumer-wcs/wx-consumer-wcs-all.jar
$SOAPUI_HOME/bin/ext
```

( ) 14. Launch SoapUI *from the* WCS validation directory.

```
cd $WCS_VALIDATION/soapUI/validation;
$SOAPUI_HOME/bin/soapui.sh
```

From the **File** menu in SoapUI, select **Import Project** and import the WCS SoapUI project file, WCSRI-112-soapui-project.xml, from the validation directory as follows:

```
File -> Import Project ->
$WCS_VALIDATION/soapUI/validation/src/test/resources/soapui/wcs112/installVal
idation/WCSRI-112-soapui-project.xml
```

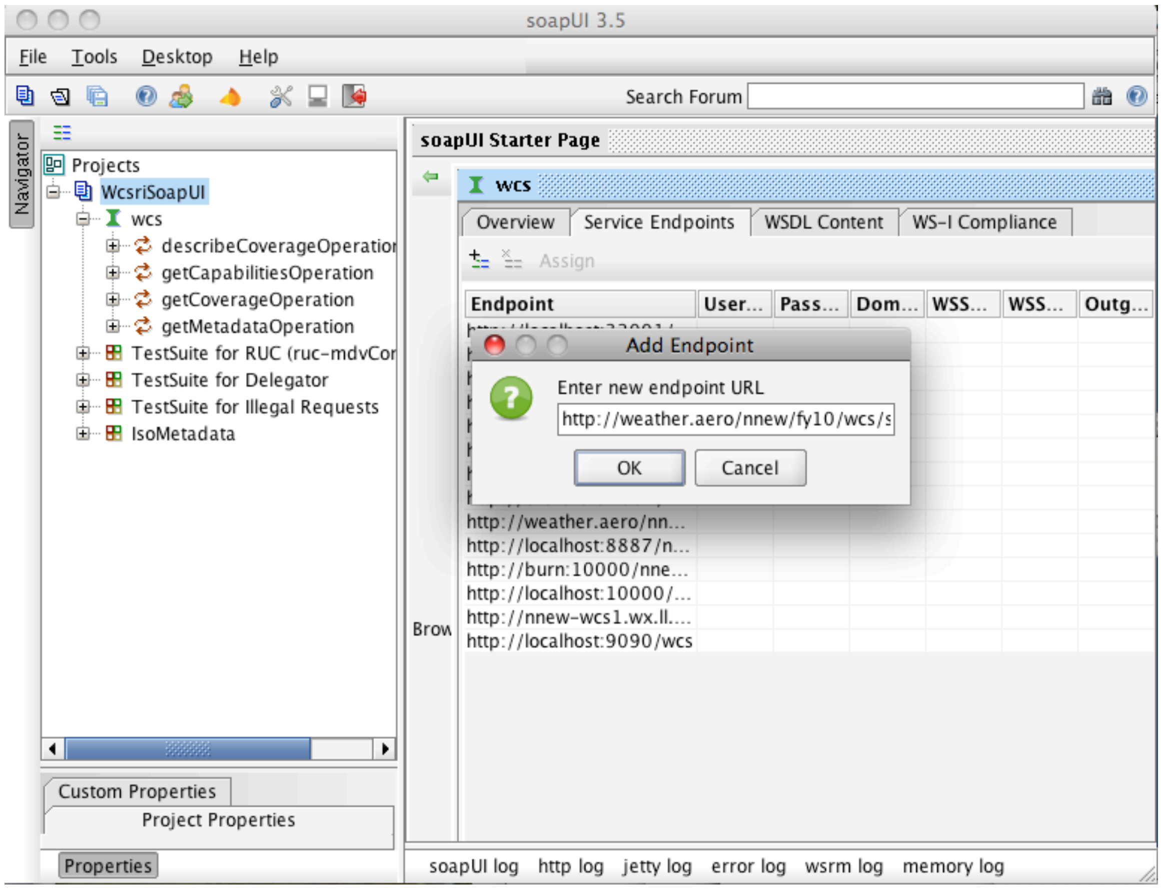


( ) 15. After the project has been imported, double click on the entry named "**wcs**" with the green hourglass. This should load the WSDL definition and show overview information on the right pane.

( ) 16. Click on the "Service Endpoints" tab.

( ) 17. Select the endpoint that corresponds to your `wcs.frontend.external.url` property (eg: http://wcs.wjhtc.nnew:8280/wcs/soap) or add another endpoint as follows.

Add the endpoint for your WCSRI installation to the tests if necessary. From the right hand pane, select "**Service Endpoints**" tab. Click on the "**+**" icon on the top left corner of this pane. In the dialog-box titled "**Add Endpoint**", add the value of the `wcs.frontend.external.url` property (eg: http://wcs.wjhtc.nnew:8280/wcs/soap) from the $FUSE_ESB_HOME/etc/wcsri.cfg file. An example is shown below, but note that your endpoint URL will be different.



Add Endpoint

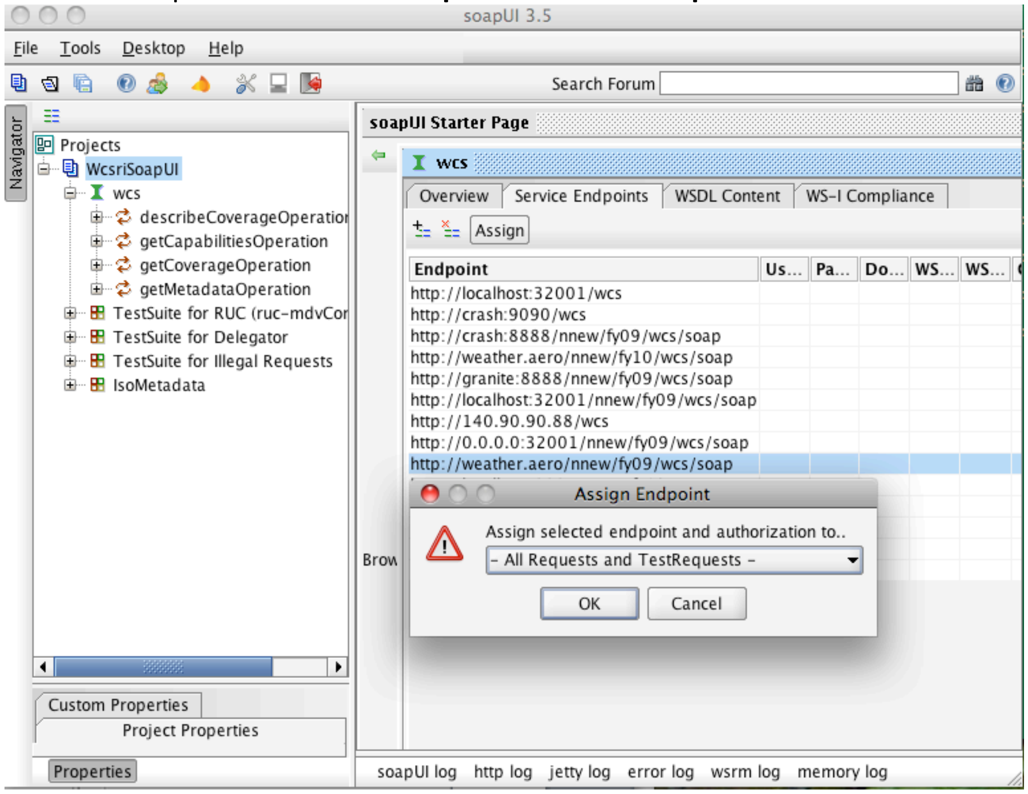
Enter new endpoint URL

http://weather.aero/nnew/fy10/wcs/s

OK Cancel

Endpoint
http://weather.aero/nn...
http://localhost:8887/n...
http://burn:10000/nne...
http://localhost:10000/...
http://nnew-wcs1.wx.ll....
http://localhost:9090/wcs

( ) 18. Select the newly added endpoint by left clicking on it and click on the "**Assign**" Button on the top-left. Select "**All Requests and Test Requests**" as shown below:



Endpoint	Us...	Pa...	Do...	WS...	WS...
http://localhost:32001/wcs					
http://crash:9090/wcs					
http://crash:8888/nnew/fy09/wcs/soap					
http://weather.aero/nnew/fy10/wcs/soap					
http://granite:8888/nnew/fy09/wcs/soap					
http://localhost:32001/nnew/fy09/wcs/soap					
http://140.90.90.88/wcs					
http://0.0.0.0:32001/nnew/fy09/wcs/soap					
http://weather.aero/nnew/fy09/wcs/soap					

Assign Endpoint

Assign selected endpoint and authorization to..

- All Requests and TestRequests -

OK Cancel

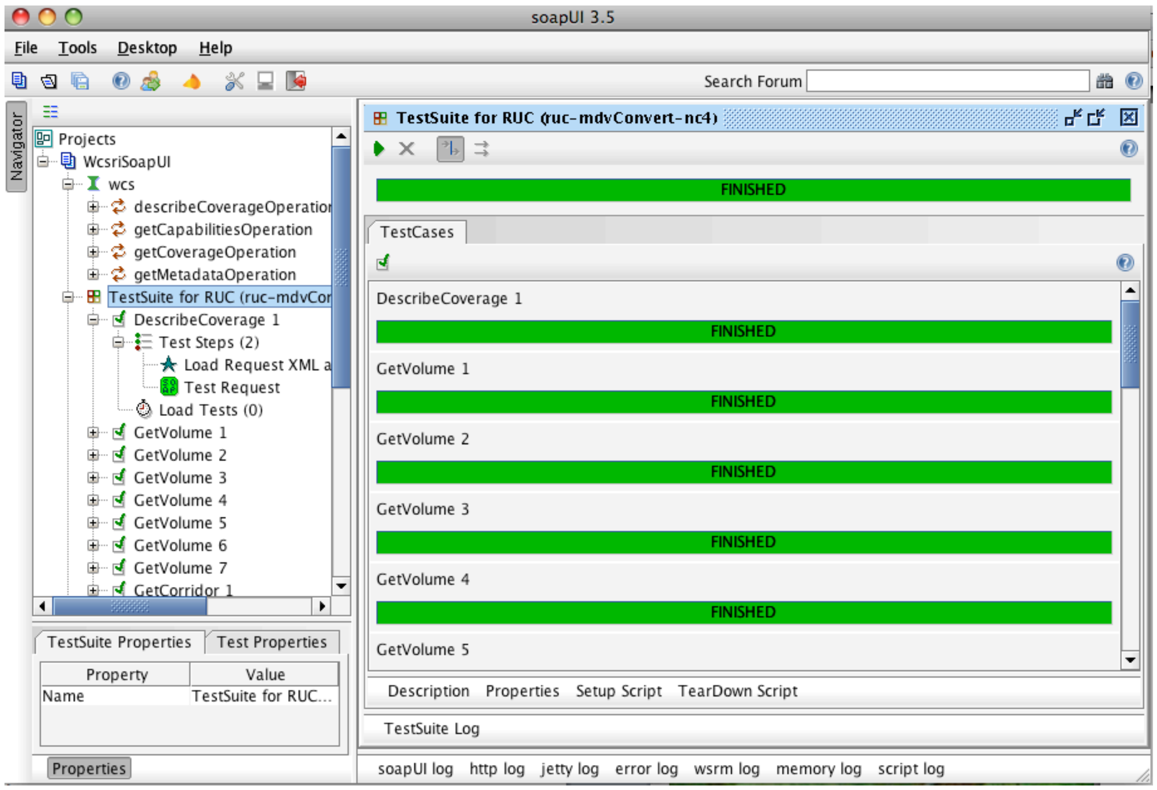
Execute Tests in SoapUI and perform a DescribeCoverage operation:

In the procedures above, we configured SoapUI to talk to the server endpoint. The following steps will describe running the WCSRI test-suite. The test suite illustrates (via execution and response assertions) all of the requirements listed above, either directly or indirectly.

( ) 19. Double click on "**Test Suite for RUC (ruc-mdvConvert-nc4)**". This will open up test cases in the right pane. Now, click on the "**Play**" button which will run the tests as shown below.

All tests should run successfully and appear green. Note that the Soap tests are a series of regression tests that validate the correct functionality of the server. The tests verify that:

- requests are issued
- responses are received
- the XML content of the responses are correct.
- if Netcdf data is returned, the data is saved and validated by dimensionality and (partially) data values.



( ) 20. Optional: Verify the "GetCapabilities 1" request/response (Req 5.3.2 and 5.3.3). On the left, under the "**Test Suite for RUC (ruc-mdvConvert-nc4)**" node, open the **"GetCapabilities 1"** node and double click on the green "**Test Request**". A new window will open displaying the request and response XML. View the response XML on the right hand side and verify that the `<ProviderName>` element has the correct content per the change made in the servicesMeta.xml file as optionally made in Step 5. For example:

```
<ows:ProviderName>WJHTC</ows:ProviderName>
```

Note that the namespace (eg: `ows`) may be different.

( ) 21. Optional: Verify the "DescribeCoverage 1" request/response (Req 5.5.1). On the left, under the "**Test Suite for RUC (ruc-mdvConvert-nc4)**" node, open the **"DescribeCoverage 1"** node and double click on the green "**Test Request"**. A new window will open displaying the request and response XML. View the response XML on the right hand side and verify that the `<Title>` element has the correct content per the change made in the coverageMeta1.xml file as optionally made in Step 6. For example:

```
<ows:Title>WJHTC Test Dataset - RUC</ows:Title>
```

Note that the namespace (eg: `ows`) may be different. Also review the content for the `<Field>` element for "TMP" (ie: Temperature), and note that it supports a "nearest"-neighbor interpolation method, has units of measure (ie: UOM) and additional metadata (Req 5.4.4.1, 5.6.2.2, 5.6.2.3).

( ) 22. Optional: Verify the "**GetVolume 1**" request/response by viewing the resulting data file (Req 4.2.1.1, 4.2.3.2, 4.3.2, 5.4.4.2, 5.7.1). On the left, under the "**Test Suite for RUC (ruc-mdvConvert-nc4)**" node, open the **"GetVolume 1"** node and double click on the green "**Test Request"**. A new window will open displaying the request and response XML. View the response XML on the right hand side and save the attachment data file.

Either:

a) View the data's dimensionality and the data itself using `ncdump`:

```
$NC4_HOME/bin/ncdump <filename> | more
```

Note that `ncdump` will show that the data has a `z0` and `time` dimensionality of 1, as requested.

b) Or start ToolsUI (on the Demo machine) and then click on the "**FeatureTypes"** tab and then the "**Grids"** tab. Open the file saved above, click on the "Temperature" grid and then the **Red** strange looking button. When a new window opens, again, click on the **Red** strange looking button. A plot of the data returned from the "GetVolume 1" request above will be displayed.

( ) 23. Optional: Verify the "GetCorridor 1" request/response by dumping the resulting data file (Req 4.2.4.1, 4.2.5.1, 4.3.2). On the left, under the "**Test Suite for RUC (ruc-mdvConvert-nc4)**" node, open the **"GetCorridor 1"** node and double click on the green "**Test Request"**. A new window will open displaying the request and response XML. View the response XML on the right hand side and save the attachment data file. View the data's dimensionality and the data itself using `ncdump`:

```
$NC4_HOME/bin/ncdump <filename> | more
```

Note that `ncdump` will show that the data has the following dimensionality:

z_dim = 21 ;
y_dim = 101 ;
x_dim = 500 ;

( ) 24. Run the "**IsoMetadata"** Test Suite by double clicking on it from the left-hand pane and then clicking on the green "**Play"** button (Req 5.9.1). All tests should run successfully and appear green.

This metadata differs from metadata returned by `GetCapabilities` and `DescribeCoverage` requests in that the "IsoMetadata" is ISO-19139 compliant. The ISO-19139 metadata is modified statically, by the Service Provider, and is not dynamically generated. Retrieving ISO metadata data is done using a new operation available through the WCSRI's WSDL, called "`GetMetadata`".

Optional: View the *services-level* ISO metadata by opening the "**Services Meta 1**" test case, double clicking on the "**Test Request**" and viewing the response XML. The content of the response comes directly from the following file (where the optional file name is specified in the wcsri.cfg file):
`[wcs.rootDataDir]/iso-metadata-service-NCAR-WCS-01.xml`, which is edited by the Service Provider.

Optional: View the *coverage-level* ISO metadata by opening the "**Services Meta 1**" test case, double clicking on the "**Test Request**" and viewing the response XML. The content of the response comes directly from a file optionally specified for each coverage, and edited by the Service Provider.

4.4.2 Test Case 2: Validate Key/Value Pair HTTP requests

Purpose and Scope:
This test case will validate that the key/value pair (KVP) HTTP requests work as expected.

Reference Documents:
doc/INSTALLATION.txt

Test Description:
In this test, we will send Key-Value Pair requests to the WCSRI server.

System Under Test:
WCSRI version 2.1

Test Setup:
- Note that this Test Case must be run directly following Test Case 1.
- ServiceMix should still be running from the previous Test Case 1.
- The next test case requires that SoapUI is setup as described in Test Case 1 – keep it running.

Test Equipment:
- WCSRI installed on the WCS Server.
- Browser (preferably FireFox)

Personnel:
Tests conducted by test engineers at FAA Tech Center. NCAR support provided by wcsri-support@rap.ucar.edu

Location:
FAA Tech Center

Test Conduct:

Safety Considerations:
See error log files available at /tmp/wcsri.log

Requirements under test:

Number	Brief Description
5.3.1	The WCSRI shall support Key-Value Pair (KVP) encoding over http GET for the GetCapabilities operation
5.3.2	Support for SOAP GetCapabilities.
5.3.3	Support Sections parameters.
5.5.1	Support for SOAP DescribeCoverage.
5.6.2.2	Provide support for spatial interpolation method of “nearest neighbor”
5.6.2.3	Provide Units of measure for each field.

5.7.1	Support for SOAP GetCoverage.
5.9.1	Support for ISO-19139 GetMetadata operation.
7.2	Provide WSDL definition for the WCSRI.

Verify WSDL and specific Requirements:

( ) 1. On the Demo machine, start Firefox, and in the location bar enter the URL for the `wcs.frontend.external.url` property (as defined in $FUSE_ESB_HOME/etc/wcsri.cfg on the WCS server) followed by "?wsdl". For example:

```
http://wcs.wjhtc.nnew:8280/wcs/soap?wsdl
```

Executing that will retrieve the WSDL from the WCSRI server (Req 7.2, 5.3.2, 5.5.1, 5.7.1).

( ) 2. Verify that XML was returned within the browser. Note that the WSDL is too complex to describe here, though you may scroll to the bottom of the screen, review the `<service name="wcsService">` element and note that it has a port binding. Also, note that there is a new WCSRI operation, called "`getMetadataOperation`" that returns (optionally) ISO 19139 metadata from the server (Req 5.9.1).

Verify Key-Value Pair (KVP) Request/Reply:

( ) 3. In the Firefox location bar, enter the `wcs.frontend.external.url` property (as defined in $FUSE_ESB_HOME/etc/wcsri.cfg on the WCS server) followed by:

```
"?service=WCS&version=1.1.2&request=GetCapabilities"
```

For example:

```
http://wcs.wjhtc.nnew:8280/wcs/soap?service=WCS&version=1.1.2&request=GetCapa
bilities
```

Executing that will perform a KVP request for `GetCapabilities` (Req 5.3.1).

( ) 4. Verify that XML was returned within the browser. The response XML should show XML nearly identical to the XML as returned from a `GetCapabilities` SOAP request.

Optional: Verify that the XML in the browser is nearly identical to the XML returned from the SoapUI test used in Test Case 1. In SoapUI, on the left under the "**Test Suite for RUC (ruc-mdvConvert-nc4)**" node, open the **"GetCapabilities 1"** node and double click on the green "**Test Request**". A new window will open displaying the request and response XML. Visually compare the response XML on the right hand side against the XML shown in Firefox, and verify that the `<ProviderName>` element has the correct content per the change made in the servicesMeta.xml file as optionally made in Test Case 1 Step 5. For example:

```
<ows:ProviderName>WJHTC</ows:ProviderName>
```

( ) 5. In the Firefox location bar, append “&sections=ServiceIdentification” to the URL used in step 3. For example:

```
http://wcs.wjhtc.nnew:8280/wcs/soap?service=WCS&version=1.1.2&request=GetCapa
bilities&sections=ServiceIdentification
```

Executing that will perform a KVP request for `GetCapabilities` while only requesting the “ServiceIdentification” section rather than the entire GetCapabilities document (Req 5.3.3).

( ) 6. In the Firefox location bar, enter the `wcs.frontend.external.url` property (in $FUSE_ESB_HOME/etc/wcsri.cfg) followed by:

```
“?service=WCS&version=1.1.2&request=DescribeCoverage&identifiers=urn:fdc:ncar
.ucar.edu:Dataset:PoxDixonNetcdf4RUC20_Air_Temperature”
```

For example:

```
http://wcs.wjhtc.nnew:8280/wcs/soap?service=WCS&version=1.1.2&request=Describ
eCoverage&identifiers=urn:fdc:ncar.ucar.edu:Dataset:PoxDixonNetcdf4RUC20_Air_
Temperature
```

Executing that will perform a KVP request for `DescribeCoverage`.

( ) 7. Verify that XML was returned within the browser. The response XML should show XML nearly identical to the XML as returned from an equivalent (ie: the same coverage identifier) `DescribeCoverage` SOAP request.

Optional: Verify that the XML in the browser is nearly identical to the XML returned from the SoapUI test used in Test Case 1. In SoapUI, on the left under the “**Test Suite for RUC (ruc-mdvConvert-nc4)**” node, open the **“DescribeCoverage 1”** node and double click on the green “**Test Request”**. A new window will open displaying the request and response XML. Visually compare the response XML on the right hand side against the XML shown in Firefox, and verify that the `<Title>` element has the correct content per the change made in the coverageMeta1.xml file as optionally made in Test Case 1 Step 6. For example:

```
<ows:Title>WJHTC Test Dataset - RUC</ows:Title>
```

( ) 8. Shutdown servicemix. From the servicemix prompt on the WCS server, type **shutdown**. This may not work due to a known bug in servicemix. If that is the case, “nicely” kill servicemix by performing the following from a terminal window on the “wcs” machine:

```
ps -ef | grep servicemix
kill [the process number of servicemix]
```

Note that “kill -9” may be required. If that’s the case, then prior to proceeding to the next test case, remove the data, deploy, and pubsub_data directories in $FUSE_ESB_HOME, and then note that you will have to reinstall the “wcsri” feature in Test Case 3 – Step 10.

4.4.3 Test Case 3: Validate Distribution WCSRI Server as a Repeater

Purpose and Scope:
Setup a distribution server as a repeater, for a single specific coverage, to an origin server running at NCAR.

Reference Documents:
doc/INSTALLATION.txt

Test Description:
In this test, we will setup a distribution server locally, which will act as a repeater for a single dataset or coverage whose origin is a WCSRI server at NCAR.

System Under Test:
WCSRI version 2.1

Test Setup:

- Note that this Test Case must be run directly following Test Case 2.
- This test case requires that SoapUI is setup as described in Test Case 2 – keep it running.
- FTI network to NCAR network tunnel.

Test Equipment:
WCSRI installed on the WCS Server.
SoapUI on the Demo machine.

Personnel:
Tests conducted by test engineers at FAA Tech Center. NCAR support provided by wcsri-support@rap.ucar.edu

Location:
FAA Tech Center (Distribution Server), NCAR (Origin Server)

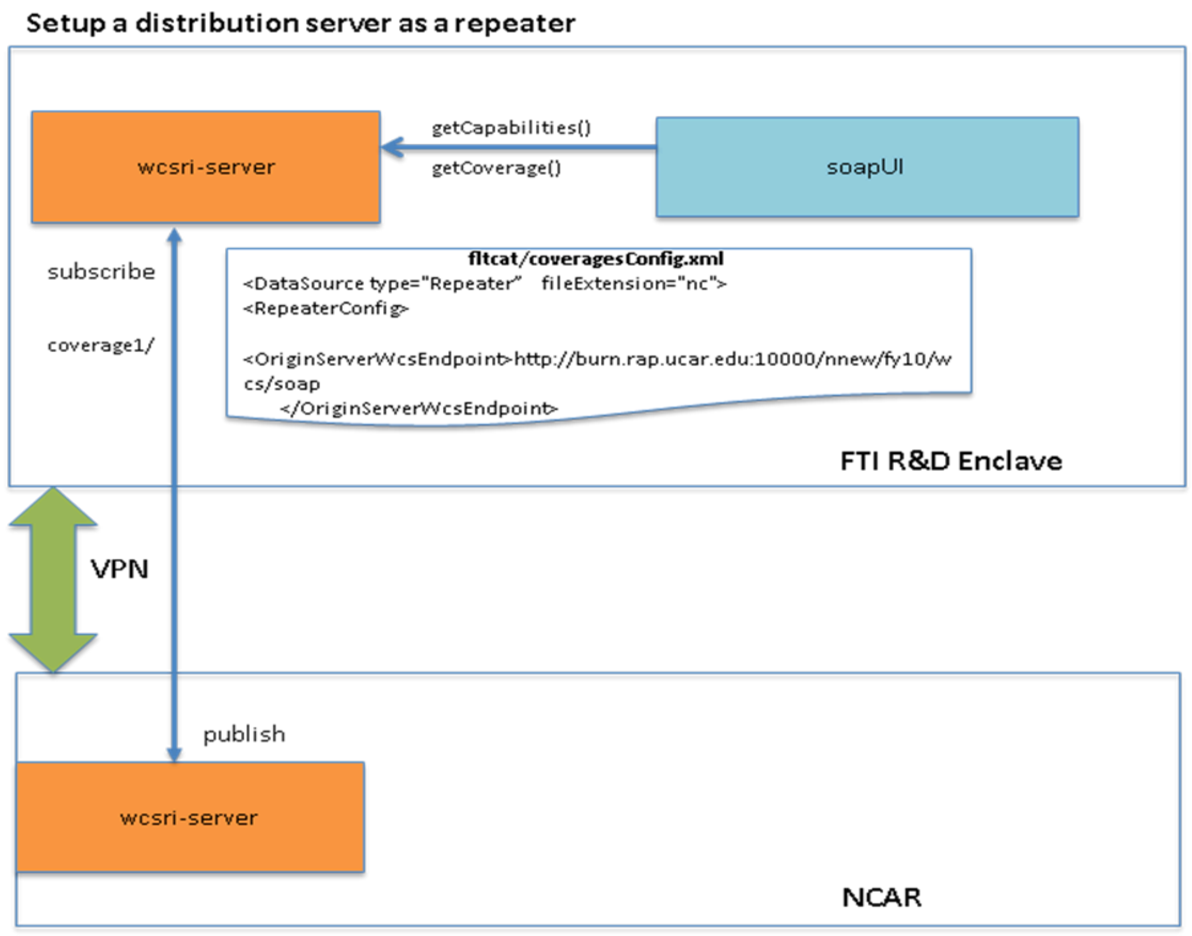
Test Conduct:

Safety Considerations:
See error log files available at /tmp/wcsri.log

Requirements under test:

Number	Brief Description
4.1.1	Provide ability for the Service Provider to specify the available coverages that will be served by the WCSRI.
4.1.2	Provide ability to configure id, name, directory and metadata for each coverage.

4.1.5	Provide lifecycle management, namely scrubbing.
4.2.1	Provide subsetting capability by field.
4.2.1.1	Provide geometric subsetting of coverage volumes.
4.2.3.1	Provide a list of valid times representing data availability.
4.2.3.2	Provide temporal subsetting by a single valid time.
4.2.3.3	Constrain a coverage by an analysis time and valid time combination (e.g., retrieve a coverage subset that was generated as part of the 12:00Z model run cycle, and is valid for 14:00Z)
4.3.1	Support CF-Netcdf4 as a native file format.
4.3.2	Expose CF-Netcdf4 at the protocol level.
6.1.1	Provide a way to distribute filtered data to interested consumers (repeater).
6.2.1	WCSRI shall be capable of acting as a data consumer of WCSRI services.
6.2.6	The WCSRI shall be capable of acting as a consumer of one or more upstream WCSRIs using their data push mechanism to cache data locally.
6.2.7	Support for configuring architectural patterns.


Setup a distribution server as a repeater
wcsri-server
getCapabilities()
getCoverage()
soapUI
subscribe
coverage1/
fltcat/coveragesConfig.xml
<DataSource type="Repeater" fileExtension="nc">
<RepeaterConfig>
<OriginServerWcsEndpoint>http://burn.rap.ucar.edu:10000/nnew/fy10/wcs/soap
</OriginServerWcsEndpoint>
FTI R&D Enclave
VPN
publish
wcsri-server
NCAR

Test Procedures:

( ) 1. Log on to the WCS server if you have not already done that.

ssh wcsri@wcs

( ) 2. From the last test case, verify that servicemix is stopped on the WCS server.

```
ps -ef | grep servicemix
```

If not, then revisit the last step of the previous test case. Remove any old log files:

```
rm /tmp/wcsri.log
```

( ) 3. Verify that the NCAR Origin Server is reachable by opening Firefox from the Demo machine, and trying the following URL:

```
http://weather.aero/nnew/fy10/wcs/soap?wsdl
```

This should result in a XML response showing the WSDL. There is a known issue with the XML Gateway such that "?" characters are problematic. If you encounter that issue, remove the "soap?" from the above URL, as in:

```
http://weather.aero/nnew/fy10/wcs/wsdl
```

Note - if this does not work, then it means there are networking issues between NCAR and FTI tunnel.

( ) 4. On the WCS server, setup a new, clean data directory which will hold data files repeated from NCAR. This will subsequently be referred to as $NEW_DATA_DIR. For example:

```
mkdir /home/users/wcsri/capEvalFY10/newDataDir
```

Create a directory for the repeated flight category data:

```
mkdir -p $NEW_DATA_DIR/fltcat
```

( ) 5. Edit $FUSE_ESB_HOME/etc/wcsri.cfg to have a new data root directory of $NEW_DATA_DIR. For example:

```
wcs.rootDataDir=/home/users/wcsri/capEvalFY10/newDataDir
```

( ) 6. Copy servicesMeta.xml and ISO metadata from the validation test dataset to the directory specified by `wcs.rootDataDir`, as in:

```
cp $WCS_VALIDATION/testData/servicesMeta.xml $NEW_DATA_DIR
cp $WCS_VALIDATION/testData/iso-metadata-service-NCAR-WCS-01.xml $NEW_DATA_DIR
```

Review the files if you wish.

( ) 7. Either create a new coveragesConfig.xml file in the $NEW_DATA_DIR/fltcat directory (Req 4.1.1, 4.1.5, 4.2.3.1, 4.3.2, 6.2.7), as below, or copy the coveragesConfig.xml file as follows:

```
cp
/home/users/wcsri/capEvalFY10/wcsriConfig/NCAR_Origin/fltcat/coveragesConfig.xml
$NEW_DATA_DIR/fltcat
```

Review the contents of $NEW_DATA_DIR/fltcat/coveragesConfig.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
  <DataSource type="Repeater" fileExtension="nc">
    <RepeaterConfig>
      <OriginServerWcsEndpoint>
         http://weather.aero/nnew/fy10/wcs/soap
      </OriginServerWcsEndpoint>
      <User></User>
    </RepeaterConfig>
    <ValidTimesType type="TimeList"/>
    <Janitor enabled="false">
      <ScrubFrequency cronPattern="0 0 2 ? * *"/>
      <KeepTime duration="-P15D"/>
    </Janitor>
    <Coverage>
      <Id>urn:fdc:ncar.ucar.edu:Dataset:FLTCAT</Id>
      <CoverageMetadataFile name="coverageMeta1.xml"/>
      <Field name="Flight_Category" id=" Flight_Category"/>
    </Coverage>
  </DataSource>
```

Notice that the XML specifies the following:

- the `DataSource` is of type `Repeater` (Req 6.2.7)
- the origin server endpoint is pointing to NCAR's weather.aero server
- the `Id` of the repeated coverage is "… FLTCAT" referring to the Flight Category dataset.
- The repeated FLTCAT coverage will contain the "Flight_Category" field.

When the distribution WCSRI starts up, it will find this "Repeater" data source and use the root directory for caching that DataSource's data. The distribution server will send a subscription request (including parameters such as coverage Id and field name) to the WCSRI running on the origin server, and will receive an unique id for its subscription. The distribution server will then connect to the origin server over JMS, sending the unique id, and will keep that connection open. When the origin server finds data pertinent to the distribution server's subscription parameters, it will "push" notifications in real-time. Each time the distribution server is notified, it will request the appropriate data and store it locally such that it can be served up from its local file system (Req 6.1.1, 6.2.1, 6.2.6). Also note that notifications and subsequent file or GetCoverage requests include *both valid and generation time* – hence, GetCoverage operations can now be executed against a WCSRI server including both bits of information (Req 4.2.3.2, 4.2.3.3).

( ) 8. Either create a new coverageMeta1.xml file (Req 4.1.2) in the $NEW_DATA_DIR/fltcat directory, as below, or copy the coverageMeta1.xml file as follows:

```
cp
/home/users/wcsri/capEvalFY10/wcsriConfig/NCAR_Origin/fltcat/coverageMeta1.xml
$NEW_DATA_DIR/fltcat
```

Contents of $NEW_DATA_DIR/fltcat/coverageMeta1.xml:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<wcs:CoverageDescriptions
   xmlns:wcs="http://www.opengis.net/wcs/1.1"
   xmlns:gml="http://www.opengis.net/gml"
   xmlns:ows="http://www.opengis.net/ows/1.1"
   xmlns:xlink="http://www.w3.org/1999/xlink">
   <wcs:CoverageDescription>
         <ows:Title>NCAR Repeater FLTCAT</ows:Title>
         <ows:Abstract>NCAR Repeater FLTCAT</ows:Abstract>
         <ows:Keywords>
               <ows:Keyword>FLTCAT</ows:Keyword>
               <ows:Keyword>Repeater</ows:Keyword>
         </ows:Keywords>
         <ows:Metadata xlink:type="simple"
                       about="http://weather.aero"/>
   </wcs:CoverageDescription>
</wcs:CoverageDescriptions>
```

( ) 9. Start servicemix by issuing the following command:

```
cd $FUSE_ESB_HOME
./bin/servicemix
```

Wait for approximately 2 minutes.

( ) 10. **If and only if** the $FUSE_ESB_HOME's data, deploy and pubsub_data directories were removed at the end of the last test case, reinstall the wcsri feature as follows, at the servicemix prompt:

```
features:install wcsri
```

Wait for approximately 2 minutes.

( ) 11. After a few minutes, you should start seeing file received event messages in the servicemix window. The file event messages will look similar to:

```
[2010-09-05 14:41:21.194] INFO  …] NNEW-Development
edu.ucar.ral.wcsri.pubsub.datasource.poller.internal.TransformationMultiplexer -
Received a file event
```

Note that the origin data has a 5 minute frequency, and therefore, you will not receive file events any more frequently than that. After you have seen a file event logged, change directory to the data directory to verify data files are being transferred from the NCAR server (Req 4.2.1, 4.2.1.1).

```
cd $NEW_DATA_DIR/fltcat
cd coverage1/2010/<YYYYMMdd>/<v_HHMMSS>/
```

Optional: Verify any “repeated” Netcdf (ie: .nc) data files using the `ncdump` command as shown below:

```
$NC4_HOME/bin/ncdump <filename> | more
```

Optional: Use ToolsUI to display any of those “repeated” Netcdf files found in the $NEW_DATA_DIR/fltcat/2010 subdirectories.

Note that the last two optional steps allow you to dump or view the repeated data, which essentially becomes the *native* data to be served by the distribution server.

( ) 12. From the Demo machine, issue a `GetCapabilities` request to verify that the FLTCAT coverage is available from the distribution server. From SoapUI, open the “**Test Suite for RUC (ruc-mdvConvert-nc4)**” node, open the **“GetCapabilities 1”** node and then double click on the green “**Test Request”**. A new window will open displaying the request and (stale from Test Case 1) response XML. For this *single test request,* press the “**Play”** green button.

In the `GetCapabilities` response XML, scroll down and verify that the following coverage identifier is listed (in any “Identifier”’s “AllowedValues” list):

```
<ns:Value>urn:fdc:ncar.ucar.edu:Dataset:FLTCAT</ns:Value>
```

This indicates that the distribution server is aware of the repeated FLTCAT coverage, and capable of serving its data (if any). If the FLTCAT identifier appears in the XML response, but the test case shows red (ie: failure), that is *not* a problem.

( ) 13. Optional: From the Demo machine, issue a `DescribeCoverage` request to verify the FLTCAT coverage. From SoapUI, open the “**Test Suite for RUC (ruc-mdvConvert-nc4)**” node, open the **“DescribeCoverage 1”** node and then double click on the green “**Test Request”**. A new window will open displaying the request and (stale from Test Case 1) response XML. Within the request, change the `<Identifier>` element content to read “FLTCAT” rather than “PoxDixon….”, as in:

```
<ns:Identifier>urn:fdc:ncar.ucar.edu:Dataset:FLTCAT</ns:Identifier>
```

For this *single test request,* press the “**Play”** green button. Note that the assertions for this test case will *expectedly fail*, since we are executing the test against the FLTCAT dataset rather than the expected "ruc-mdvConvert” dataset. Note the list of `<timePosition>` elements which conveys the valid times for the list of available FLTCAT data files on the distribution server. Optionally, wait another 5 minutes and repeat the `DescribeCoverage` request, to see additional valid times.

( ) 14. Optional: Issue a GetCoverage against the FLTCAT dataset (Req 4.3.1), and ncdump the response. From SoapUI, open the “**Test Suite for RUC (ruc-mdvConvert-nc4)**” node, open the **“GetVolume 1”** node and then double click on the green “**Test Request”**. In the new window, edit the request XML per properties of the FLTCAT coverage, by a) changing the `Identifier` to "…FLTCAT" b) changing the `FieldSubset's Identifier` to "Flight_Category"c) changing the `BoundingBox's` dimensions to 2 d) removing the Z component of the `BoundingBox` and e) changing the `Begin/End timePositions` to a value copy/pasted from the results of the

( ) 15. previous `DescribeCoverage`. Press the green "**Play**" button to execute the test. Again, the assertions for this test case will *expectedly fail.*

Save the response's attachment, and ncdump the file as follows:

```
$NC4_HOME/bin/ncdump <filename> | more
```

( ) 16. Shutdown servicemix. From the servicemix prompt on the WCS server, type **shutdown**. This may not work due to a known bug in servicemix. If that is the case, "nicely" kill servicemix by performing the following from a terminal window on the "wcs" machine:

```
ps –ef | grep servicemix
kill [the process number of servicemix]
```

Note that "kill -9" may be required. If that's the case, then prior to proceeding to the next test case, remove the data, deploy, and pubsub_data directories in $FUSE_ESB_HOME, and then note that you will have to reinstall the "wcsri" feature in Test Case 4 – Step 7.

Test Data Reduction and Analysis:

In this test, we have successfully shown Repeater functionality for a WCSRI distribution server. The server points to an origin server at NCAR and caches the files locally to be treated as native files for the "distributed/repeated" coverage.

4.4.4 Test Case 4: Validate distribution WCSRI Server as a Delegator

Purpose and Scope:
In this test, we will delegate requests to the origin server at NCAR. This test will show the delegator pattern of distribution.

Reference Documents:
doc/INSTALLATION.txt

Test Description:
This test will setup a distribution server that will delegate the client requests to the origin server at NCAR. The data files will not be cached locally. The test procedures are similar to Test Case 4. We will be setting up a delegator instead of a repeater in the coverage configuration.

System Under Test:
WCSRI version 2.1

Test Setup:
- Note that this Test Case must be run directly following Test Case 3.
- This test case requires that SoapUI is setup as described in Test Case 2 – keep SoapUI running.
- FTI network to NCAR network tunnel.

Test Equipment:
WCSRI installed on the WCS Server.
SoapUI on the Demo machine.

Personnel:
Tests conducted by test engineers at FAA Tech Center. NCAR support provided by wcsri-support@rap.ucar.edu

Location:
FAA Tech Center (Distribution Server), NCAR (Origin Server)

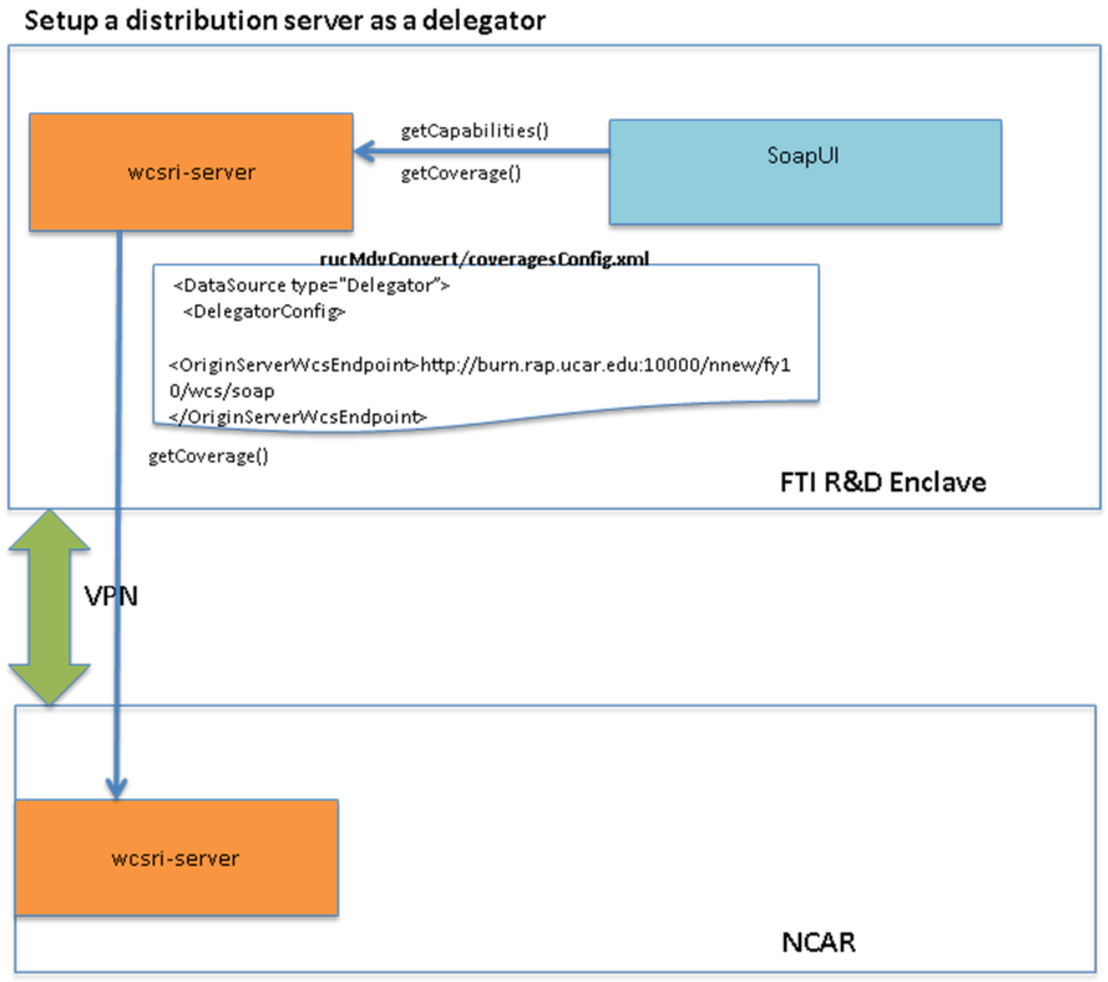
Test Conduct:

Safety Considerations:
See error log files available at /tmp/wcsri.log

Requirements under test:

Number	Brief Description
4.1.1	Provide ability for the Service Provider to specify the available coverages that will be served by the WCSRI.
6.2.1	WCSRI shall be capable of acting as a data consumer of WCSRI services.

6.2.5	The WCSRI shall be capable of acting as a proxy or request delegate to another upstream WCSRI
6.2.7	Support for configuring architectural patterns.


Setup a distribution server as a delegator
wcsri-server
getCapabilities()
getCoverage()
SoapUI
rucMdvConvert/coveragesConfig.xml
<DataSource type="Delegator">
<DelegatorConfig>
<OriginServerWcsEndpoint>http://burn.rap.ucar.edu:10000/nnew/fy10/wcs/soap</OriginServerWcsEndpoint>
getCoverage()
FTI R&D Enclave
VPN
wcsri-server
NCAR

Test Procedures:

( ) 1. Log on to the WCS server if you have not already done that.

ssh wcsri@wcs

( ) 2. From the last test case, verify that servicemix is stopped on the WCS server.

```
ps -ef | grep servicemix
```

If not, then revisit the last step of the previous test case. Remove any old log files:

```
rm /tmp/wcsri.log
```

( ) 3. Verify that the NCAR Origin Server is reachable by opening Firefox from the Demo machine, and trying the following URL:

```
http://weather.aero/nnew/fy10/wcs/soap?wsdl
```

This should result in a XML response showing the WSDL. There is a known issue with the XML Gateway such that "?" characters are problematic. If you encounter that issue, remove the "soap?" from the above URL, as in:

```
http://weather.aero/nnew/fy10/wcs/wsdl
```

Note - if this does not work, then it means there are networking issues between NCAR and FTI tunnel.

( ) 4. Create a new coverage data directory to delegate requests for the test data set called ruc-mdvConvert.

```
mkdir $NEW_DATA_DIR/rucMdvConvert;
cd $NEW_DATA_DIR/rucMdvConvert;
```

( ) 5. Create a new coveragesConfig.xml file (Req 4.1.1) in the $NEW_DATA_DIR/rucMdvConvert directory, as below, or copy the coveragesConfig.xml file as follows:

```
cp
/home/users/wcsri/capEvalFY10/wcsriConfig/NCAR_Origin/rucMdvConvert/coveragesCon
fig.xml $NEW_DATA_DIR/rucMdvConvert
```

Review the contents of $NEW_DATA_DIR/rucMdvConvert/coveragesConfig.xml:

```
<?xml version="1.0" encoding="UTF-8"?>

<DataSource type="Delegator">
  <DelegatorConfig>
    <OriginServerWcsEndpoint>
      http://weather.aero/nnew/fy10/wcs/soap
    </OriginServerWcsEndpoint>
    <User></User>
  </DelegatorConfig>
  <Coverage>
   <Id>
  urn:fdc:ncar.ucar.edu:Dataset:PoxDixonNetcdf4RUC20_Air_Temperature
   </Id>
  </Coverage>
</DataSource>
```

Notice that the XML specifies the following:

- the `DataSource` is of type `Delegator` (Req 6.2.7)
- the origin server endpoint is pointing to NCAR's weather.aero server
- the `Id` of the delegated coverage refers to the test dataset available *from the NCAR WCS server*.

When the distribution WCSRI starts up, it will find this "Delegator" data source to be used for all coverage `Id`s listed in the coveragesConfig.xml file. All `DescribeCoverage` and `GetCoverage` requests targeted for these coverages will be *delegated* to the upstream WCSRI server as specified by the `OriginServerWcsEndpoint`. (Req 6.2.1, 6.2.5). Note that the data files for delegated coverages remain on the origin server and are *not cached locally* on the distribution server.

( ) 6. Verify that the WCSRI's data directory (`wcs.rootDataDir`) is pointing to $NEW_DATA_DIR in the $FUSE_ESB_HOME/etc/wcsri.cfg file. This was the same value used in Test Case 3.

( ) 7. Start servicemix by issuing the following command from a terminal window:

```
cd $FUSE_ESB_HOME
./bin/servicemix
```

Wait for approximately 2 minutes.

If and only if the $FUSE_ESB_HOME's data, deploy and pubsub_data directories were removed at the end of the last test case, reinstall the wcsri feature as follows, at the servicemix prompt:

```
features:install wcsri
```

Wait for approximately 2 minutes.

( ) 8. Execute the "**Test Suite for RUC (ruc-mdvConvert-nc4)**" tests by double clicking on the "**Test Suite for RUC (ruc-mdvConvert-nc4)**" node, and then clicking the green **Play** button. All tests should succeed *except for* "GetCapabilities 1" which looks for additional coverage Ids not available from the distribution server as it's currently set up.

Note that from Test Case 1, the endpoint used for the tests is the value of your `wcs.frontend.external.url` property (`http://wcs.wjhtc.nnew:8280/wcs/soap`). Hence, the ruc-mdvConvert tests execute against the endpoint for the distribution server. However, note that the WCS server as set up in Step 6 above uses the $NEW_DATA_DIR as the WCSRI's data root, and does *not actually have* this data directly. Instead, all DescribeCoverage and GetCoverage requests for the test data set are delegated to NCAR's origin server.

( ) 9. List the files in the data directory (eg: $NEW_DATA_DIR/rucMdvConvert/) and note that there are no data files that accumulate there since all requests are delegated to the NCAR server.

4.5 WCSRI 2.0 Requirements Coverage Summary

Reqmt #	Requirement Description	WCSRI impl. begin version	WCSRI impl. finish version	Test Case#
4.1.1	The WCSRI shall provide the capability for the Service Provider to specify the available coverages or datasets that will be served by the WCSRI.	V1	V2	1, 3, 4
4.1.2	The WCSRI shall provide the capability for the Service Provider to configure the following for each dataset offered by the WCSRI: • A unique identifier for the coverage. • Name of the coverage. • The directory for the coverage data files. • Descriptive metadata for the dataset.	V1	V2	1, 3
4.1.5	The WCSRI shall provide a lifecycle management system that will delete data beyond a certain time period or archive data for a specified period of time	V1	V2	1, 3
4.2.1	The WCSRI shall allow subsetting by zero or more fields (e.g., temperature, reflectivity). If no fields are indicated, all available fields shall be returned	V1	V1	1, 3
4.2.1.1	The WCSRI shall support geometric subsetting of coverage volumes, including the following: • 3-D Volume • Horizontal 2-D Cross-section • Vertical 2-D Cross-section • Sounding • 1-D Point • Trajectory • Corridor	V1	V3	1, 3
4.2.3.1	The WCSRI shall be capable of providing a list of valid	V1	V1	1, 3

	times from datasets that are temporally aggregated			
4.2.3.2	The WCSRI shall provide the capability to constrain coverages by requesting a single valid time.	V1	V1	1, 3
4.2.3.3	The WCSRI shall provide the capability to constrain a coverage by an analysis time and valid time combination (e.g., retrieve a coverage subset that was generated as part of the 12:00Z model run cycle, and is valid for 14:00Z)	V2	V2	1, 3
4.2.4.1	The WCSRI shall provide the capability to aggregate coverages over valid time resulting in a time series of coverages for a given set of constraints (e.g., geometry of temperature)	V1	V3	1
4.2.5.1	The WCSRI shall support temporal trajectories for trajectory-related geometries (e.g., trajectory, corridor, etc)	V1	V2	1
4.3.1	The WCSRI shall support CF-NetCDF4 as a native file format	V1	V1	1, 3
4.3.2	The WCSRI shall expose CF-NetCDF4 as a file format at the protocol level	V1	V1	1, 3
4.3.3	The WCSRI shall support GRIB2 as a native file format	V1	V1	1
5.2.1	The WCSRI shall provide a means for the Service Provider to enable/disable the WCS operations by encoding type	V1	V2	1
5.3.1	The WCSRI shall support Key-Value Pair (KVP) encoding over http GET for the GetCapabilities operation	V1	V2	2
5.3.2	For the GetCapabilities operation, the WCSRI shall support SOAP encoding over HTTP POST	V1	V1	1, 2
5.3.3	The WCSRI shall support the Sections parameters	V1	V2	1, 2
5.4.1.1	The WCSRI shall provide a means for the Service Provider to statically configure metadata describing the server for the following GetCapabilities elements: ServiceIdentification, ServiceProvider, OperationsMetadata and Contents	V1	V2	1
5.4.2.1	The WCSRI shall provide a means for the Service Provider to statically configure the endpoints for the GetCapabilities, DescribeCoverage and GetCoverage web services, for KVP, SOAP	V1	V2	1
5.4.3.1	The WCSRI shall provide a means for the Service Provider to configure the path for storing temporary files and scrubbing instructions (e.g., maximum file age) for stored request results and other temporary file purposes	V1	V2	1

5.4.4.1	The WCSRI shall provide the capability to determine the following, on a per-coverage basis, either through programmatic means or Service Provider configuration: 5 Unique Coverage Identifier 6 World Geodetic System 1984 (WGS84) Bounding Box for the coverage 7 Supported coordinate reference systems (CRS), including geographic projections, vertical and compound. 8 Native CRS	V1	V2	1
5.4.4.2	The WCSRI shall support “application/x-NetCDF” as the value for the `supportedFormat` attribute. This is the Multipurpose Internet Mail Extensions (MIME) type for CF-NetCDF4 binary coverage data	V1	V1	1
5.5.1	For the DescribeCoverage operation, the WCSRI shall support SOAP encoding over HTTP POST	V1	V1	1, 2
5.6.2.1	The WCSRI shall provide a means for the Service Provider to configure metadata for each of the fields within each offered coverage	V1	V2	1
5.6.2.2	The WCSRI shall provide a capability to support a spatial interpolation method of “nearest neighbor”	V1	V2	1, 2
5.6.2.3	The WCSRI shall provide Units of Measure for each field within a coverage as part of the DescribeCoverage results	V1	V2	1, 2
5.7.1	For the GetCoverage operation, the WCSRI shall support SOAP encoding over HTTP POST	V1	V1	1, 2
5.9.1	The WCSRI shall support a GetMetadata operation. The level of metadata requested (e.g., dataset, field), as well as the particular metadata content, shall be specified in the operation request	V2	V2	1, 2
6.1	The WCSRI installation shall include, at a minimum, a description of the procedure for adding and removing offered coverages, and updating the metadata for those coverages	V1	V3	1
6.1.1	The WCSRI shall provide a low-latency means to distribute filtered data (such as geometric and temporal subsets) to interested data consumers. The mechanism used to distribute filtered data shall be consistent with what is used by other 4-D Wx Data Cube fundamental components, such as the WFSRI	V2	V2	3

6.2.1	The WCSRI shall be capable of acting as a data consumer of WCSRI services	V2	V3	3,4
6.2.5	The WCSRI shall be capable of acting as a proxy or request delegate to another upstream WCSRI	V2	V2	4
6.2.6	The WCSRI shall be capable of acting as a consumer of one or more upstream WCSRIs using their data push mechanism to cache data locally	V2	V2	3
6.2.7	The WCSRI shall provide the capability to configure a variety of architectural patterns on a product-by-product basis (i.e., different coverages may warrant different WCSRI communication patterns)	V2	V2	1, 3,4
7.2	The WCSRI shall include WSDL definitions of the web services that it provides	V1	V3	2
10.1	The WCSRI shall be capable of running a self-test diagnostic following installation. Canned datasets and default configurations will be provided with the software release for that purpose	V1	V3	1

5 Implementation Verification - Registry Repository

5.1 Implementation Verification – Registry / Repository

The NNEW Registry/Repository ("RegRep" or simply "Registry") is a virtual information catalog used to store high-level information (i.e., metadata) about both datasets and the associated services that provide those datasets to users. Each service's interface information is used at build-time to generate client applications conforming to those interfaces. At run-time, meanwhile, the Registry's metadata allows users to discover datasets and determine which service(s) are capable of providing that data. Thus, users can search for concepts like *air_temperature*, determine which datasets contain references to those concepts (and which such datasets are classified as members of the NextGen Single Authoritative Source—the SAS), and discover the services storing that data, all via the Registry's interface.

The NNEW Registry is federated—it consists of a group of individual Registry instances, each operated independently by a different NNEW member organization. The interoperability between the different registries in the federation is enabled by the OASIS ebXML RegRep 4 specifications.

The test cases in this chapter cover both the core build-time and run-time usage scenarios for the NNEW Registry/Repository.

5.2 Test Environment and Setup

5.2.1 wxForge Projects

The setup of the various NNEW registries and the tests run against them requires a set of configuration files, the contents of the registries, and the custom software used to interact with them. This data and software has been checked into the wxForge subversion repositories hosted by MIT Lincoln Laboratory. Before you configure the registries described in this document, or run any tests on them, all their related projects must be checked out. A subversion client is required.

- **Wx Data Cube Consumer Library** – The MIT-LL RegRep client library is required to perform some of the test queries on the registries. This library is available from the NCAR wxconsumer project. Use the following command to check the wxconsumer project out to the current working directory, substituting `<user>` with a valid account name. The directory you check it out to will be referred to throughout this chapter as `$WXCONSUMER`.

```
svn co --username <user> http://wxForge.wx.ll.mit.edu/svn/wxconsumer
```

- **Ontologies** – A set of ontology and alignment files from the Ontologies project must be loaded in order to perform semantic searches within the registries. Use the following command to check the Ontologies project out to the current working directory, substituting

`<user>` with a valid account name. The directory you check it out to will be referred to throughout this chapter as `$ONTOLOGIES`.

```
svn co --username <user> http://wxForge.wx.ll.mit.edu/svn/ontologies
```

- **Weather Cube Metadata** – The test datasets, services, taxonomies, and configuration files for the registries are available from the wxcube-metadata project. Use the following command to check the wxcube-metadata project out to the current working directory, substituting `<user>` with a valid account name. The directory you check it out to will be referred to throughout this chapter as `$TEST_DATA_DIR`.

```
svn co --username <user>
   http://wxForge.wx.ll.mit.edu/svn/wxcube-metadata
```

5.2.2 Client Software

The tests defined in this chapter use the following registry client software:

1. Java-based Registry Administration UI – A UI that enables publishing and discovery of datasets, service instances, service interfaces, taxonomies, and more. Its discovery features allow for searching through individual registries with local searches, or through multiple-registry federations with federated searches.

2. Registry Test Client command line program – A test program used to test the fault-tolerance and client-side support for federated queries within the regrep4-client API library.

5.2.3 Registry and Federation Setup

The 2010 Test Plan defines the following registry federation setup:

1. NNEW Federation 1 – A federation with the following Registry/Repositories:

- FAA Technology Center 1 – A registry actively operated by the William J. Hughes Technical Center, otherwise known as the FAA Tech Center. Abbreviated throughout this document as “WJHTC Registry.”

- Lincoln Labs Registry 1 – A registry actively operated by MIT Lincoln Labs. Abbreviated as “MIT-LL Registry.”

- NWS Registry 1 – A registry actively operated by the National Weather Service. Abbreviated as “NWS Registry.”

- GSD Registry 1 – A registry actively operated by NOAA/GSD. Abbreviated as “GSD

Registry."

The exact configuration for the above federation is available as wjhtc-federation-rim.xml.

5.2.4 Initial Data Setup

At the start of the test, each registry is pre-loaded with the following types of information:

1. **Dataset descriptions** for the datasets provided by a given registry's operating organization. Each description consists of a file formatted according to the ISO 19139 metadata standard—a worldwide standard slated to replace the FGDC standard in the U.S. in the years ahead.

2. **Service descriptions** for the services provided by the given registry's operating organization. Service information is specified as a combination of ISO 19139 metadata and W3C WSDL files.

3. **Taxonomies and other configuration data** for each registry. These include any taxonomies, object types, association types, and other configuration data required by the NNEW Registry profile—for example, the weather cube taxonomy. These pre-loaded objects are defined using the ebXML RegRep 4 Registry Information Model schema.

4. **Ontology and Alignment data** for allowing semantic searches of datasets in the registry.

Note that all test metadata is defined in the wxcube-metadata or ontologies projects on wxForge. The hyperlinks above link to organization-specific directories within the project.

5.2.5 Launching the Registry Administration UI version 4.7

Most of the tests in this chapter will use the Registry Administration UI version 4.7. It can be launched via one of the following methods:

- Point your web browser to the following URL:

  http://regrep-primary.wjhtc.nnew:8090/wellgeo-ui-swing/4.7/jnlp/wellgeo-ui-swing.jnlp

- Use the `javaws` program as follows:

```
javaws wellgeo-ui-swing.jnlp
```

This will launch an instance of the Registry Administration UI version 4.7 with the various NNEW registries configured. By default, the Admin UI will connect to the WJHTC Registry. To select another registry (e.g. MIT-LL), you may select its URL in the **Registry Base URL** field of the **Options** dialog, accessible via **Tools → Options** on the menu bar. Note that switching the Registry Base URL may take up to a minute, with no busy indicator shown.

5.2.6 Registry Configuration

The following are directions for setting up each registry with the respective datasets, services, and other configurations required for the tests that follow. The administrator for each of the registries should follow the directions in their corresponding section and in the "All Registries" section to prepare for the tests.

- WJHTC

  1. Open the Registry Administration UI **version 4.7** for the WJHTC registry.
  2. Log in to the Registry using a valid User ID and Password.
  3. Go to **Tools → Wizards → Import Directory Tree**
  4. Click the **Choose** button and open the directory `$TEST_DATA_DIR/trunk/src/main/resources/FY10-demo/wjhtc`
  5. Click the **Next** button. In the next window mark the check boxes for:
     - …`/wjhtc`
     - …`/datasets`
     - …`/services`
  6. Click **Finish**. You should see confirmation of the upload.

- MIT-LL

  1. Open the Registry Administration UI **version 4.7** for the MIT-LL registry.
  2. Log in to the Registry using a valid User ID and Password.
  3. Go to **Tools → Wizards → Import Directory Tree**
  4. Click the **Choose** button and open the directory `$TEST_DATA_DIR/trunk/src/main/resources/FY10-demo/mit-ll`
  5. Click the **Next** button. In the next window mark the check boxes for:
     - …`/mit-ll`
     - …`/datasets`

- …/services

6. Click **Finish**. You should see confirmation of the upload.

7. Go to **Tools → Wizards → Import ebRIM File**

8. Click the **Choose** button and open the file `$TEST_DATA_DIR/trunk/src/main/resources/mit-ll/accessControlPolicies/mit-RoleConfig-rim.xml`

9. Click **Finish**. You should see a confirmation of the upload.

10. If you already have the regrep4-client library installed, then change the working directory to `$WXCONSUMER/trunk/regrep/util`. If not, do the following:

    a) Change the working directory to `$WXCONSUMER/trunk/`
    b) Run "`mvn install`".
    c) Change the working directory to `$WXCONSUMER/trunk/regrep/util`

11. Run "`mvn compile appassembler:assemble`"

12. Change directory to `$WXCONSUMER/trunk/regrep/util/target/bin`

13. Run "`chmod 755 *`"

14. Type the following command to run the `pub19139` utility (replace `<username>` and `<password>` with a valid username and password for a member of the MIT organization group). It is recommended that you copy and paste the command into the terminal, given its length.

```
./pub19139 -v -noconfirm -url http://nnew-
   regrep1.wx.ll.mit.edu/omar-server -u <username> -p <password> -a
   urn:test:ll.mit.edu-ACP $TEST_DATA_DIR/trunk/wxcube-
   metadata/src/main/resources/datasets/test/iso-metadata-dataset-
   Product.xml $TEST_DATA_DIR/trunk/wxcube-
   metadata/src/main/resources/datasets/test/iso-metadata-dataset-Product-
   R.xml $TEST_DATA_DIR/trunk/wxcube-
   metadata/src/main/resources/datasets/test/iso-metadata-dataset-Product-
   C.xml $TEST_DATA_DIR/trunk/wxcube-
   metadata/src/main/resources/datasets/test/iso-metadata-dataset-Product-
   S.xml $TEST_DATA_DIR/trunk/wxcube-
   metadata/src/main/resources/datasets/test/iso-metadata-dataset-Product-
   TS.xml $TEST_DATA_DIR/trunk/wxcube-
   metadata/src/main/resources/services/test/iso-metadata-service-MIT-WFS-
   Test.xml $TEST_DATA_DIR/trunk/wxcube-
   metadata/src/main/resources/services/test/MIT-WFS-Test.wsdl
```

The current version of the Registry does not apply the security constraints to all of the RegistryObjects. To allow for some of the following tests the XML for the RegistryObjects must be edited manually.

15. Launch the Registry Admin UI **version 4.7**.

16. Login as the Registry administrator using the login controls in upper-right corner of the screen.

17. Click on the **Search** tab in upper-left corner of the UI to access the Search Tool panel.

18. Make sure that the **Federated Query Options** combo box shows **Local Query** as its selection.

19. Select **Get RegistryObject By ID** in the **Select Query** combo box.

20. In the ID text field, enter “urn:fdc:ll.mit.edu:Service:WFS-Test:ServiceEndpoint:wfs-SOAP-Port” and then click **Search**.

21. Right-click on the match and select **Open**. In the editor pane, click on the **XML** button. Edit the text field to contain the following **XML** block. Copy and paste only the boldfaced XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RegistryObject xsi:type="ServiceEndpointType"
serviceBinding="http://www.opengis.net/wfs/soap:ServiceBinding:wfs-SOAP"
address="http://ngen-wfsri.wx.ll.mit.edu/soap/wfs"
status="urn:oasis:names:tc:ebxml-regrep:StatusType:Submitted"
owner="regadmin" objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:ServiceEndpoint"
lid="urn:fdc:ll.mit.edu:Service:WFS-Test:ServiceEndpoint:wfs-SOAP-Port"
id="urn:fdc:ll.mit.edu:Service:WFS-Test:ServiceEndpoint:wfs-SOAP-Port"
xmlns:ns2="http://www.w3.org/1999/xlink" xmlns="urn:oasis:names:tc:ebxml-
regrep:xsd:rim:4.0" xmlns:ns4="urn:oasis:names:tc:ebxml-
regrep:xsd:query:4.0" xmlns:ns3="http://www.w3.org/2005/08/addressing"
xmlns:ns5="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:4.0"
xmlns:ns6="urn:oasis:names:tc:ebxml-regrep:xsd:spi:4.0"
xmlns:ns7="urn:oasis:names:tc:ebxml-regrep:xsd:rs:4.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Slot name="urn:oasis:names:tc:ebxml-
      regrep:rim:RegistryObject:accessControlPolicy">
        <ValueList>
            <ValueListItem xsi:type="StringValueType">
                <Value>urn:test:ll.mit.edu-ACP</Value>
            </ValueListItem>
        </ValueList>
    </Slot>
    <Slot collectionType="urn:oasis:names:tc:ebxml-
      regrep:CollectionType:List"
      name="urn:iso:TC211:19115:slot:metadataConstraints:MD_LegalConstrain
      ts:accessConstraints">
        <ValueList>
            <ValueListItem xsi:type="VocabularyTermValueType">
                <Value term="restricted"
                  vocabulary="http://www.isotc211.org/2005/resources/Codel
                  ist/gmxCodelists.xml#MD_RestrictionCode"/>
            </ValueListItem>
        </ValueList>
    </Slot>
    <Name>
        <LocalizedString value="wfs-SOAP-Port" xml:lang="en-US"/>
    </Name>
    <Description>
        <LocalizedString value="" xml:lang="en-US"/>
    </Description>
    <VersionInfo userVersionName=""/>
</RegistryObject>
```

22. Click the **Update Model** button in the lower-right corner of the Editor Pane. Right-click in the Editor Pane and then select **Save**.

23. Repeat Step 20 but use the following ID:

urn:fdc:ll.mit.edu:Service:WFS-Test:Service:MIT_WebFeatureService-Test

24. Repeat Step 21 but use the following XML:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RegistryObject xsi:type="ServiceType" status="urn:oasis:names:tc:ebxml-
regrep:StatusType:Submitted" owner="mitPerson1"
objectType="urn:oasis:names:tc:ebxml-
regrep:ObjectType:RegistryObject:Service"
lid="urn:fdc:ll.mit.edu:Service:WFS-Test:Service:MIT_WebFeatureService-
Test" id="urn:fdc:ll.mit.edu:Service:WFS-
Test:Service:MIT_WebFeatureService-Test"
xmlns:ns2="http://www.w3.org/1999/xlink" xmlns="urn:oasis:names:tc:ebxml-
regrep:xsd:rim:4.0" xmlns:ns4="urn:oasis:names:tc:ebxml-
regrep:xsd:query:4.0" xmlns:ns3="http://www.w3.org/2005/08/addressing"
xmlns:ns5="urn:oasis:names:tc:ebxml-regrep:xsd:lcm:4.0"
xmlns:ns6="urn:oasis:names:tc:ebxml-regrep:xsd:spi:4.0"
xmlns:ns7="urn:oasis:names:tc:ebxml-regrep:xsd:rs:4.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <Slot name="urn:oasis:names:tc:ebxml-
      regrep:rim:RegistryObject:accessControlPolicy">
        <ValueList>
            <ValueListItem xsi:type="StringValueType">
                <Value>urn:test:ll.mit.edu-ACP</Value>
            </ValueListItem>
        </ValueList>
    </Slot>
    <Slot collectionType="urn:oasis:names:tc:ebxml-
      regrep:CollectionType:List"
      name="urn:iso:TC211:19115:slot:metadataConstraints:MD_LegalConstrain
      ts:accessConstraints">
        <ValueList>
            <ValueListItem xsi:type="VocabularyTermValueType">
                <Value term="restricted"
                  vocabulary="http://www.isotc211.org/2005/resources/Codel
                  ist/gmxCodelists.xml#MD_RestrictionCode"/>
            </ValueListItem>
        </ValueList>
    </Slot>
```

```
        <Name>
            <LocalizedString value="MIT_WebFeatureService-Test" xml:lang="en-
              US"/>
        </Name>
        <Description>
            <LocalizedString value="MIT Web Feature Service instance. Supports
              HTTP interface" xml:lang="en-US"/>
        </Description>
        <VersionInfo userVersionName=""
          versionName="6b5511022f615d032b19750a22c40b200aa11930"/>
        <ServiceEndpoint
          serviceBinding="http://www.opengis.net/wfs/soap:ServiceBinding:wfs-
          SOAP" address="http://ngen-wfsri.wx.ll.mit.edu/soap/wfs"
          status="urn:oasis:names:tc:ebxml-regrep:StatusType:Submitted"
          owner="regadmin" objectType="urn:oasis:names:tc:ebxml-
          regrep:ObjectType:RegistryObject:ServiceEndpoint"
          lid="urn:fdc:ll.mit.edu:Service:WFS-Test:ServiceEndpoint:wfs-SOAP-
          Port" id="urn:fdc:ll.mit.edu:Service:WFS-Test:ServiceEndpoint:wfs-
          SOAP-Port">
            <Slot name="urn:oasis:names:tc:ebxml-
              regrep:rim:RegistryObject:accessControlPolicy">
                <ValueList>
                    <ValueListItem xsi:type="StringValueType">
                        <Value>urn:test:ll.mit.edu-ACP</Value>
                    </ValueListItem>
                </ValueList>
            </Slot>
            <Slot collectionType="urn:oasis:names:tc:ebxml-
              regrep:CollectionType:List"
              name="urn:iso:TC211:19115:slot:metadataConstraints:MD_LegalConst
              raints:accessConstraints">
                <ValueList>
                    <ValueListItem xsi:type="VocabularyTermValueType">
                        <Value term="restricted"
                          vocabulary="http://www.isotc211.org/2005/resources/C
                          odelist/gmxCodelists.xml#MD_RestrictionCode"/>
                    </ValueListItem>
                </ValueList>
            </Slot>
            <Name>
                <LocalizedString value="wfs-SOAP-Port" xml:lang="en-US"/>
            </Name>
            <Description>
                <LocalizedString value="" xml:lang="en-US"/>
            </Description>
            <VersionInfo userVersionName=""/>
        </ServiceEndpoint>
    </RegistryObject>
```

- NWS

1. Open the Registry Administration UI **version 4.7** for the NWS registry.

2. Log in to the Registry using a valid User ID and Password.

3. Go to **Tools → Wizards → Import Directory Tree**

4. Click the **Choose** button and open the directory `$TEST_DATA_DIR/trunk/src/main/resources/FY10-demo/nws`

5. Click the **Next** button. In the next window mark the check boxes for:

   - …`/nws`
   - …`/datasets`
   - …`/services`

6. Click **Finish**. You should see confirmation of the upload.

- GSD

  1. Open the Registry Administration UI **version 4.7** for the GSD registry.

  2. Log in to the Registry using a valid User ID and Password.

  3. Go to **Tools → Wizards → Import Directory Tree**

  4. Click the **Choose** button and open the directory `$TEST_DATA_DIR/trunk/src/main/resources/FY10-demo/gsd`

  5. Click the **Next** button. In the next window mark the check boxes for:

     - …`/gsd`
     - …`/datasets`
     - …`/services`

  6. Click **Finish**. You should see confirmation of the upload.

- All Registries

  1) Open the Registry Administration UI **version 4.7**.

  2) Log in to the Registry using a valid User ID and Password.

  3) Go to **Tools → Wizards → Import Directory Tree**

  4) Click the **Choose** button and open the directory `$ONTOLOGIES/trunk/src/main/resources/ontologies`

  5) Click the **Next** button. In the next window mark the check boxes for:

     - …`/ontologies`
     - …`/CF`
     - …`/JMBL`
     - …`/NNEW_Weather_Ontologies`

- …/alignment_files

6) Click **Finish**. You should see a confirmation of the upload.

7) Log in to the Registry using a valid User ID and Password.

8) Go to **Tools → Wizards → Import ebRIM File**

9) Click the **Choose** button and open the file `$TEST_DATA_DIR/trunk/src/main/resources/taxonomies/ebrim-classificationScheme-TriAgencyDataCube.xml`

10) Click **Finish**. You should see a confirmation of the upload.

11) Repeat steps 8 through 10 with the file `$TEST_DATA_DIR/trunk/src/main/resources/taxonomies/ebrim-classificationScheme-ISO19119ServicesExtended.xml`

12) If you have the regrep4-client library already installed, then change the working directory to `$WXCONSUMER/trunk/regrep/util`. If not, do the following:

- Change the working directory to `$WXCONSUMER/trunk/`
- Run "`mvn install`".
- Change the working directory to `$WXCONSUMER/trunk/regrep/util`

13) Run "`mvn compile appassembler:assemble`"

14) Change directory to `$WXCONSUMER/trunk/regrep/util/target/bin`

15) Run "`chmod 755 *`"

16) Run "`./serviceClassifier –url <url> -u <username> -p <password>`", replacing `<url>` with the URL of the registry and `<username>` and `<password>` with the same username and password used to load the datasets and services.

5.3 Discovery of Dataset by Weather Phenomenon Type

This test demonstrates the query capabilities of the Registry/Repository. In particular, it demonstrates:

1. **Unfiltered Search**: Discovery of all registered datasets without specifying filter criteria.

2. **Filtered Search**: Discovery of all registered datasets, filtered by a particular weather phenomenon type. The filter provides an exact match or regular expression match to the

specified weather phenomenon.

3. **Semantic Filtered Search**: Discovery of all registered datasets, filtered by a particular weather phenomenon type and other phenomenon *similar* to it. The filter in this case expands the specified weather phenomenon to all similar phenomena via an ontology.

4. **Federated Search**: Discovery of datasets within multiple registries in a registry federation.

5.3.1 Dataset Discovery – Unfiltered Local Search

This test performs an unfiltered search for all registered data sets in the WJHTC Registry.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Click on the **Search** tab in upper-left corner of the UI to access the **Search Tool** panel.

3. Make sure that the **Federated Query Options** combo box shows **Local Query** as its selection.

4. Select **Find Data Set** in the **Select Query** combo box.

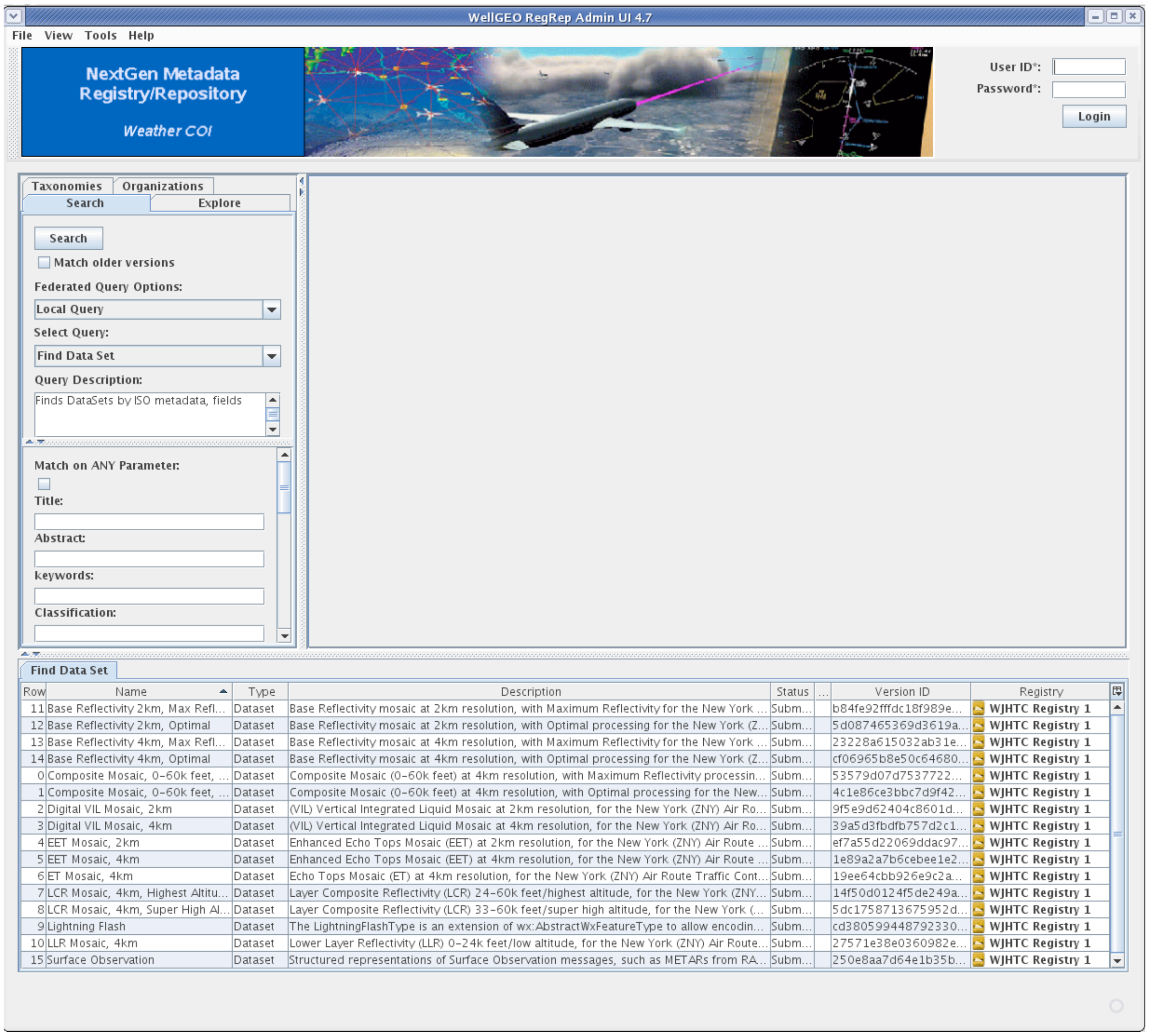
5. Click the **Search** button at the top of the **Search Tool** panel. An unfiltered list of all the registered datasets will appear. Click on the **Name** column heading to sort the datasets alphabetically by name.

6. Verify that the list of datasets returned includes the following:

   - Base Reflectivity 2km, Max Reflectivity
   - Base Reflectivity 2km, Optimal
   - Base Reflectivity 4km, Max Reflectivity

7. Right-click on the **Find Data Set** tab and select **Close**.

Note: The sorting based on name is transient and does not currently remain in effect from query to query. This has been submitted to Wellfleet as a feature request for the UI.



Federated Query Options: Local Query

Select Query: Find Data Set

Query Description: Finds DataSets by ISO metadata, fields

Row	Name	Type	Description	Status	...	Version ID	Registry
11	Base Reflectivity 2km, Max Refl...	Dataset	Base Reflectivity mosaic at 2km resolution, with Maximum Reflectivity for the New York ...	Subm...		b84fe92fffdc18f989e...	WJHTC Registry 1
12	Base Reflectivity 2km, Optimal	Dataset	Base Reflectivity mosaic at 2km resolution, with Optimal processing for the New York (Z...	Subm...		5d087465369d3619a...	WJHTC Registry 1
13	Base Reflectivity 4km, Max Refl...	Dataset	Base Reflectivity mosaic at 4km resolution, with Maximum Reflectivity for the New York ...	Subm...		23228a615032ab31e...	WJHTC Registry 1
14	Base Reflectivity 4km, Optimal	Dataset	Base Reflectivity mosaic at 4km resolution, with Optimal processing for the New York (Z...	Subm...		cf06965b8e50c64680...	WJHTC Registry 1
0	Composite Mosaic, 0-60k feet, ...	Dataset	Composite Mosaic (0-60k feet) at 4km resolution, with Maximum Reflectivity processin...	Subm...		53579d07d7537722...	WJHTC Registry 1
1	Composite Mosaic, 0-60k feet, ...	Dataset	Composite Mosaic (0-60k feet) at 4km resolution, with Optimal processing for the New...	Subm...		4c1e86ce3bbc7d9f42...	WJHTC Registry 1
2	Digital VIL Mosaic, 2km	Dataset	(VIL) Vertical Integrated Liquid Mosaic at 2km resolution, for the New York (ZNY) Air Ro...	Subm...		9f5e9d62404c8601d...	WJHTC Registry 1
3	Digital VIL Mosaic, 4km	Dataset	(VIL) Vertical Integrated Liquid Mosaic at 4km resolution, for the New York (ZNY) Air Ro...	Subm...		39a5d3fbdfb757d2c1...	WJHTC Registry 1
4	EET Mosaic, 2km	Dataset	Enhanced Echo Tops Mosaic (EET) at 2km resolution, for the New York (ZNY) Air Route ...	Subm...		ef7a55d22069ddac97...	WJHTC Registry 1
5	EET Mosaic, 4km	Dataset	Enhanced Echo Tops Mosaic (EET) at 4km resolution, for the New York (ZNY) Air Route ...	Subm...		1e89a2a7b6cebee1e2...	WJHTC Registry 1
6	ET Mosaic, 4km	Dataset	Echo Tops Mosaic (ET) at 4km resolution, for the New York (ZNY) Air Route Traffic Cont...	Subm...		19ee64cbb926e9c2a...	WJHTC Registry 1
7	LCR Mosaic, 4km, Highest Altitu...	Dataset	Layer Composite Reflectivity (LCR) 24-60k feet/highest altitude, for the New York (ZNY...	Subm...		14f50d0124f5de249a...	WJHTC Registry 1
8	LCR Mosaic, 4km, Super High Al...	Dataset	Layer Composite Reflectivity (LCR) 33-60k feet/super high altitude, for the New York (...	Subm...		5dc1758713675952d...	WJHTC Registry 1
9	Lightning Flash	Dataset	The LightningFlashType is an extension of wx:AbstractWxFeatureType to allow encodin...	Subm...		cd380599448792330...	WJHTC Registry 1
10	LLR Mosaic, 4km	Dataset	Lower Layer Reflectivity (LLR) 0-24k feet/low altitude, for the New York (ZNY) Air Route...	Subm...		27571e38e0360982e...	WJHTC Registry 1
15	Surface Observation	Dataset	Structured representations of Surface Observation messages, such as METARs from RA...	Subm...		250e8aa7d64e1b35b...	WJHTC Registry 1

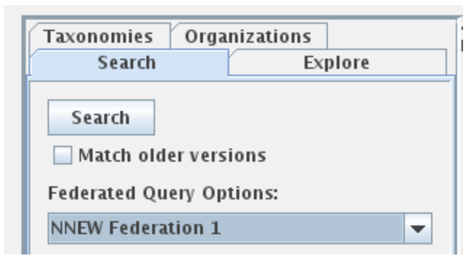
Figure 5.1: Dataset Discovery – Unfiltered Local Search
Shows datasets from WJHTC Registry

5.3.2 Dataset Discovery – Unfiltered Federated Search

This test performs an unfiltered search for all registered data sets across all registries in the NNEW Federation 1.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 .

2. Repeat the previous test, Dataset Discovery – Unfiltered Local Search, with one change: Make sure that in Step 3, the **Federated Query Options** combo box shows **NNEW Federation 1** as its selection.



Federated Query Options: NNEW Federation 1

Figure.5.2: Dataset Discovery – Unfiltered Federated Search

3. Verify that the list of datasets returned includes datasets from the GSD, MIT LL, NWS and WJHTC registries, as indicated by the last column labeled **Registry** in the search results.

4. Right-click on the **Find Data Set** tab and select **Close**.

5.3.3 Dataset Discovery – Local Search Filtered by Dataset Field

This test performs a filtered search for all data sets within the MIT-LL registry containing the weather phenomenon types *air_temperature* and *temperatureAir*, where *air_temperature* is a Climate and Forecast (CF) conventions term and *temperatureAir* is a Joint METOC Broker Language (JMBL) term.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Select the **Tools → Options** menu item. In the Registry Base URL combo box, set the URL to the MIT-LL Registry.

3. Click on the **Search** tab in upper-left corner of the UI to access the **Search Tool** panel.

4. Make sure that the **Federated Query Options** combo box shows **Local Query** as its selection.

5. Select **Find Data Set** in the **Select Query** combo box.

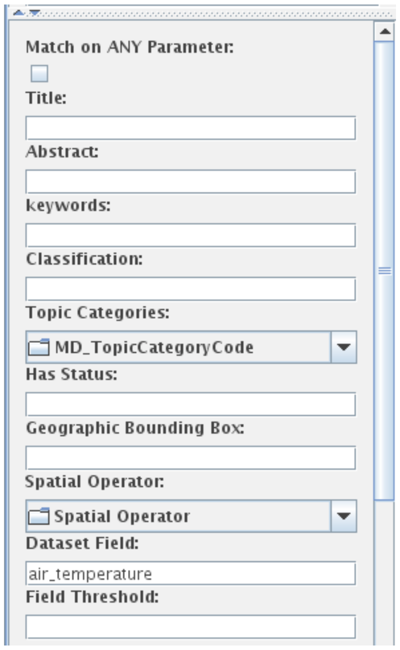
6. Type the following text in the **Dataset Field** text entry box:

   ```
   air_temperature
   ```

7. Click the **Search** button. A filtered set of all registered datasets with the weather phenomenon type *air_temperature* are returned. Click on the **Name** column heading to sort the datasets alphabetically by name.

8. Observe that the matching data sets returned include:

- GFS
  - RUC Model – 20 kilometer resolution


Match on ANY Parameter:

Title:

Abstract:

keywords:

Classification:

Topic Categories:

MD_TopicCategoryCode

Has Status:

Geographic Bounding Box:

Spatial Operator:

Spatial Operator

Dataset Field:

air_temperature

Field Threshold:

Figure 5.3: Dataset Discovery – Local Search Filtered by Dataset Field

9. Repeat the search, only this time type the following text in the **Dataset Field** text entry box:

   ```
   temperatureAir
   ```

10. Click the **Search** button. A filtered set of all registered datasets with weather phenomenon type *temperatureAir* are returned.

11. Verify that the single matching data set that is returned is:
    - DOD Model Air Temperature

12. Right-click on the **Find Data Set** tab and select **Close All**.

5.3.4 Dataset Discovery – Federated Search Filtered by Dataset Field

This test performs a filtered search for all datasets across all registries in the NNEW Federation 1 containing the weather phenomenon types *air_temperature* and *temperatureAir*, where *air_temperature* is a Climate and Forecast (CF) conventions term and *temperatureAir* is a Joint METOC Broker Language (JMBL) term.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . Make sure you are connected to the WJHTC registry by checking the **Tools → Options** menu item and confirming it is pointed to the WJHTC registry.

2. Repeat the previous test Dataset Discovery – Local Search steps 2 through 7 with one change: Make sure that in Step 3, the **Federated Query Options** combo box shows **NNEW Federation 1**

as its selection.

3. Verify that the list of datasets returned in Step 8 includes datasets from more than just the WJHTC registry as indicated by the last column labeled **Registry** in search results. It is possible that not all of the registries will have an entry to return for this query.

4. Right-click on the **Find Data Set** tab and select **Close**.

5.3.5 Dataset Discovery – Semantic Filtered Search

This test performs a semantically enhanced, filtered search for all datasets with the weather phenomenon type *temperatureAir*, a Joint METOC Broker Language term; *air_temperature,* a Climate and Forecast conventions term; and *LiquidWater*, a Joint METOC Broker Language. The Climate and Forecast term equivalent to *LiquidWater* is *atmosphere_cloud_liquid_water_content*. The test uses the Climate and Forecast ontology and Joint METOC Broker Language taxonomies, and alignments between those two entities.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.1.5. By default, it will connect with the WJHTC Registry.

2. Select the **Tools → Options** menu item. In the **Registry Base URL** combo box, set the URL to the MIT-LL Registry.

3. In the Ontology Context window, right-click and select **Check All**.

4. Click on the **Search** tab in upper-left corner of the UI to access the **Search Tool** panel.

5. Make sure that the **Federated Query Options** combo box shows **Local Query** as its selection.

6. Select **Find Data Set** in the **Select Query** combo box.

7. Type the following text in the **Dataset Field** text entry box:

   ```
   temperatureAir
   ```

8. Type the following value in the **Field Threshold** text entry box:

   ```
   0.5
   ```

   The threshold indicates that datasets whose weather phenomenon types match the specified weather phenomenon type at or above the threshold value are returned. The threshold accepts a value in the range of 0.0 to 1.0. Setting the threshold to 0.5 will pick up any datasets whose weather phenomenon types match *temperatureAir* with a 0.5 confidence or higher.

9. Click the "**Search**" button. A filtered set of all registered datasets with weather phenomenon

type of *temperatureAir* and similar (confidence > 0.5) weather phenomenon types are returned. Click on the "**Name**" column heading to sort the datasets by Name (alphabetical order).

10. Verify the matching data sets that are returned are:

- DOD Model Air Temperature
- GFS
- RUC Model – 20 kilometer resolution

11. Repeat the **Find Data Set** search (Step 6). Type the following text in the **Dataset Field** text entry box:

```
air_temperature
```

12. Type the following value in the **Field Threshold** text entry box:

```
0.5
```

13. Click the **Search** button. A filtered set of all registered datasets with the weather phenomenon type *air_temperature* and similar (confidence > 0.5) weather phenomenon types are returned. Click on the **Name** column heading to sort the datasets by Name (alphabetical order).

14. Verify that the matching data sets are the same as the ones returned in Step 9.

15. Repeat the **Find Data Set** search (Step 6). Type the following text in the **Dataset Field** text entry box:

```
LiquidWater
```

16. Leave the **Field Threshold** text entry box blank.

17. Click the **Search** button. No records are found, as this constitutes a search without ontologies.

18. Type the following value in the **Dataset Threshold** text entry box:

```
0.5
```

19. Click the **Search** button. A filtered set of all registered datasets with the weather phenomenon type *LiquidWater* and similar (confidence > 0.5) weather phenomenon types are returned.

20. Verify that the following data set is returned:

- Vertically Integrated Liquid (VIL)
- Vertically Integrated Liquid (VIL) Forecast
- VIL Contour Standard Mode Forecast

- VIL Contour Winter Mode Forecast
- VIL Standard Mode Forecast Accuracy
- VIL Winter Mode Forecast Accuracy

21. Type the following value in the **Dataset Threshold** text entry box:

    ```
    0.8
    ```

22. Click the **Search** button. No results should be returned, as there are no registered datasets with weather phenomenon type *LiquidWater* or similar (confidence > 0.8) weather phenomenon types.

23. Reset the **Dataset Field** to '*****' and clear out the **Field Threshold** for the next test to use this query.

24. Right-click on a **Find Data Set** tab and select **Close All**.

5.3.6 Dataset Discovery – REST Search Filtered By Dataset Field

This test is identical in function to the earlier Dataset Discovery – Local Search Filtered by Dataset Field test. The only difference is that this test verifies the ability to search the Registry using its REST interface, using an HTTP GET URL to initiate the search and get results in ebXML RegRep's ebRIM XML format. The REST interface is suitable for building REST clients to the Registry. To properly view the results, your web browser should be set up to view XML. In some cases, you might have to select a "view source" option for the page to see this information. It is recommended that Firefox be used for the following tests, as it displays XML data within the browser window by default.

The search URL includes the query ID, along with query parameters as URL parameters, as shown in the template URL below:

```
<server base url>/search?queryId=<the query id>&{<param-name>=<param-
value>}*&format=<application/xml | application/json>
```

Applying this to our dataset discovery query, filtered by Dataset Field, results in the following URL for the WJHTC Registry:

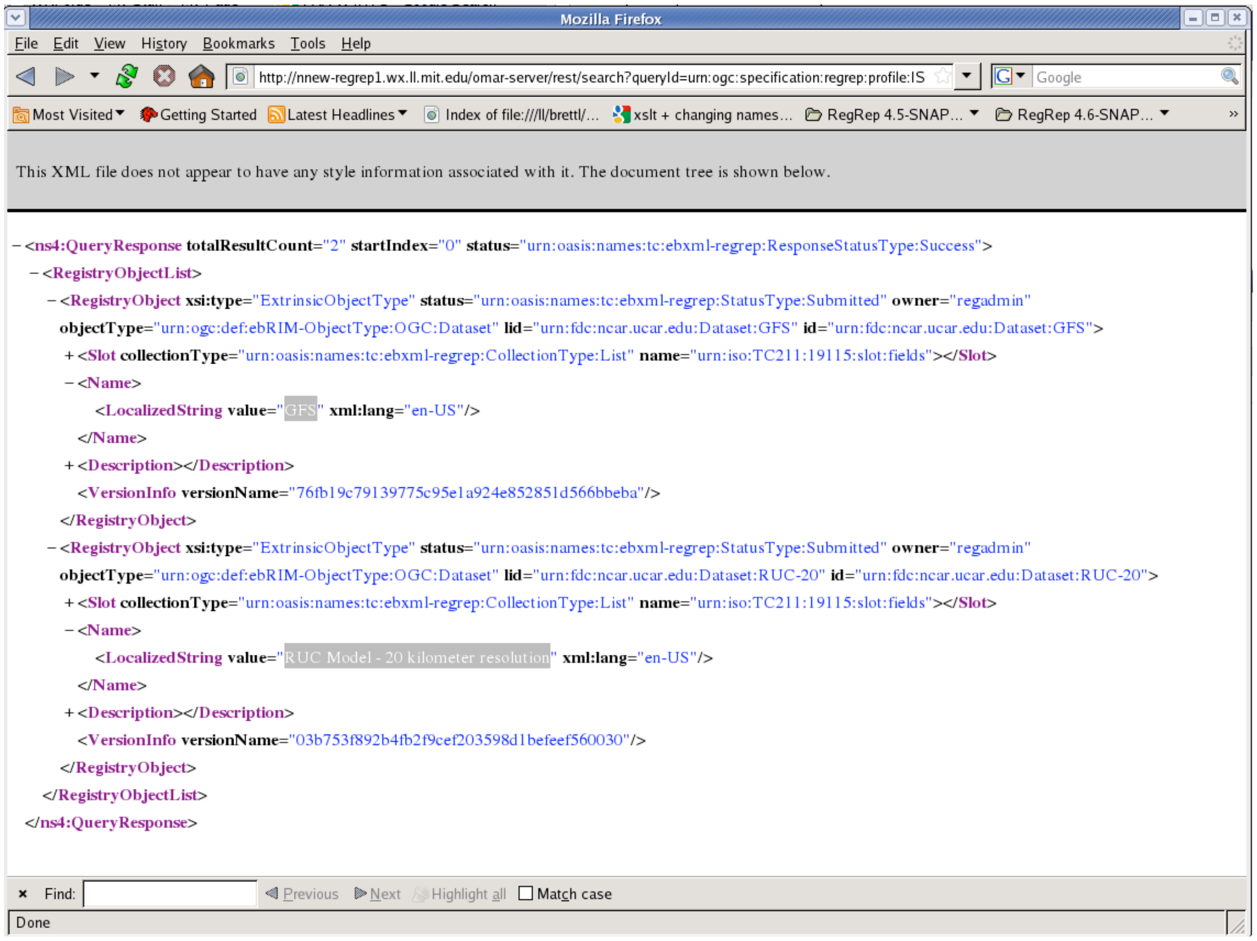
http://nnew-regrep1.wx.ll.mit.edu/omar-server/rest/search?queryId=urn:ogc:specification:regrep:profile:ISO19139:query:DatasetDiscoveryQuery&field=air_temperature&format=application/xml

1. Click on the above URL to open it in a web browser.

2. Verify that the matching data sets that are returned include:

    - GFS

- RUC Model – 20 kilometer resolution



Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://nnew-regrep1.wx.ll.mit.edu/omar-server/rest/search?queryId=urn:ogc:specification:regrep:profile:IS

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <ns4:QueryResponse totalResultCount="2" startIndex="0" status="urn:oasis:names:tc:ebxml-regrep:ResponseStatusType:Success">
  – <RegistryObjectList>
    – <RegistryObject xsi:type="ExtrinsicObjectType" status="urn:oasis:names:tc:ebxml-regrep:StatusType:Submitted" owner="regadmin"
      objectType="urn:ogc:def:ebRIM-ObjectType:OGC:Dataset" lid="urn:fdc:ncar.ucar.edu:Dataset:GFS" id="urn:fdc:ncar.ucar.edu:Dataset:GFS">
      + <Slot collectionType="urn:oasis:names:tc:ebxml-regrep:CollectionType:List" name="urn:iso:TC211:19115:slot:fields"></Slot>
      – <Name>
          <LocalizedString value="GFS" xml:lang="en-US"/>
        </Name>
      + <Description></Description>
        <VersionInfo versionName="76fb19c79139775c95e1a924e852851d566bbeba"/>
      </RegistryObject>
    – <RegistryObject xsi:type="ExtrinsicObjectType" status="urn:oasis:names:tc:ebxml-regrep:StatusType:Submitted" owner="regadmin"
      objectType="urn:ogc:def:ebRIM-ObjectType:OGC:Dataset" lid="urn:fdc:ncar.ucar.edu:Dataset:RUC-20" id="urn:fdc:ncar.ucar.edu:Dataset:RUC-20">
      + <Slot collectionType="urn:oasis:names:tc:ebxml-regrep:CollectionType:List" name="urn:iso:TC211:19115:slot:fields"></Slot>
      – <Name>
          <LocalizedString value="RUC Model - 20 kilometer resolution" xml:lang="en-US"/>
        </Name>
      + <Description></Description>
        <VersionInfo versionName="03b753f892b4fb2f9cef203598d1befeef560030"/>
      </RegistryObject>
    </RegistryObjectList>
  </ns4:QueryResponse>
```

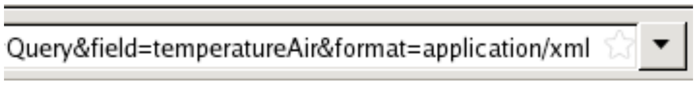
Find: Previous Next Highlight all Match case

Done

Figure 5.4: Dataset Discovery – REST Search Filtered by Dataset Field = air_temperature

3. Change the URL to specify the field name `temperatureAir` using the following URL:

http://nnew-regrep1.wx.ll.mit.edu/omar-server/rest/search?queryId=urn:ogc:specification:regrep:profile:ISO19139:query:DatasetDiscoveryQuery&field=temperatureAir&format=application/xml



Query&field=temperatureAir&format=application/xml

Figure 5.5: Dataset Discovery – REST Search by Dataset Field = temperatureAir

4. Click on the above URL to open it in a web browser.

5. Verify that the matching data sets that are returned include:

   1. DOD Model Air Temperature

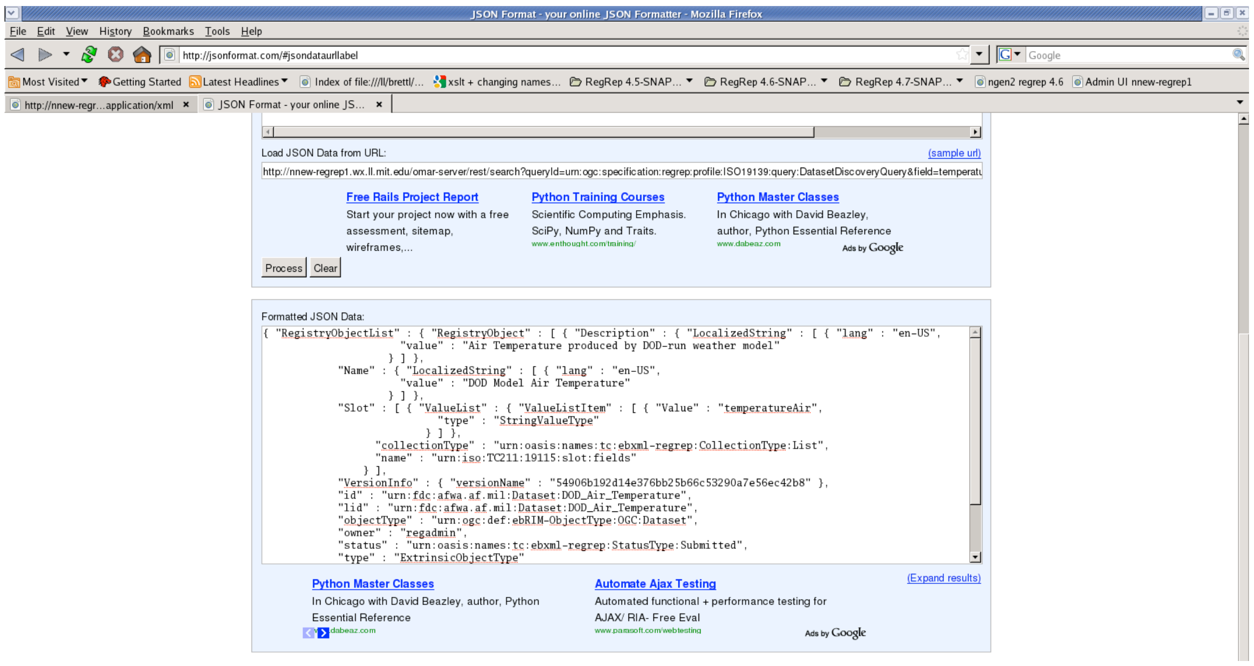
6. Perform the previous REST query, but specify `format=application/json` instead of `format=application/xml` and verify that the result is returned in JSON format.

http://nnew-regrep1.wx.ll.mit.edu/omar-server/rest/search?queryId=urn:ogc:specification:regrep:profile:ISO19139:query:DatasetDiscoveryQuery&field=temperatureAir&format=application/json

7. Click on the above URL and save the file as `json.txt`. Copy the `json.txt` file to a computer with Internet access.

8. Verify that the browser returns a JSON-formatted document representing the DOD Model Air Temperature dataset. If you wish to see the document in a cleaner display, specify above URL in the website below:

http://jsonformat.com/


JSON Format - your online JSON Formatter - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://jsonformat.com/#jsondataurllabel
Google
Most Visited
Getting Started
Latest Headlines
Index of file:///ll/brettl/...
xslt + changing names...
RegRep 4.5-SNAP...
RegRep 4.6-SNAP...
RegRep 4.7-SNAP...
ngen2 regrep 4.6
Admin UI nnew-regrep1
http://nnew-regr...application/xml
JSON Format - your online JS...
Load JSON Data from URL:
(sample url)
Free Rails Project Report
Start your project now with a free assessment, sitemap, wireframes,...
Python Training Courses
Scientific Computing Emphasis. SciPy, NumPy and Traits.
www.enthought.com/training/
Python Master Classes
In Chicago with David Beazley, author, Python Essential Reference
www.dabeaz.com
Ads by Google
Process
Clear
Formatted JSON Data:
Python Master Classes
In Chicago with David Beazley, author, Python Essential Reference
dabeaz.com
Automate Ajax Testing
Automated functional + performance testing for AJAX/ RIA- Free Eval
www.parasoft.com/webtesting
Ads by Google
(Expand results)

```
http://nnew-regrep1.wx.ll.mit.edu/omar-server/rest/search?queryId=urn:ogc:specification:regrep:profile:ISO19139:query:DatasetDiscoveryQuery&field=temperatu
```

```
{ "RegistryObjectList" : { "RegistryObject" : [ { "Description" : { "LocalizedString" : [ { "lang" : "en-US",
                      "value" : "Air Temperature produced by DOD-run weather model"
                    } ] },
            "Name" : { "LocalizedString" : [ { "lang" : "en-US",
                      "value" : "DOD Model Air Temperature"
                    } ] },
            "Slot" : [ { "ValueList" : { "ValueListItem" : [ { "Value" : "temperatureAir",
                            "type" : "StringValueType"
                          } ] },
                  "collectionType" : "urn:oasis:names:tc:ebxml-regrep:CollectionType:List",
                  "name" : "urn:iso:TC211:19115:slot:fields"
                } ],
            "VersionInfo" : { "versionName" : "54906b192d14e376bb25b66c53290a7e56ec42b8" },
            "id" : "urn:fdc:afwa.af.mil:Dataset:DOD_Air_Temperature",
            "lid" : "urn:fdc:afwa.af.mil:Dataset:DOD_Air_Temperature",
            "objectType" : "urn:ogc:def:ebRIM-ObjectType:OGC:Dataset",
            "owner" : "regadmin",
            "status" : "urn:oasis:names:tc:ebxml-regrep:StatusType:Submitted",
            "type" : "ExtrinsicObjectType"
```

Figure 5.6: Dataset Discovery – REST Search with Results in JSON Format

5.4 Discovery of Datasets by Weather Cube Domain Classification

The 4-D Wx Data Cube is partitioned into multiple domains and sub-domains. One critical sub-domain is the Single Authoritative Source (SAS), referred to often in the high-level Cube CONOPS documents. This test demonstrates browsing for datasets by weather domain capabilities in the Registry/Repository. In particular, it demonstrates:

1. **Viewing an existing taxonomy** using the Registry Admin UI.

2. **Unfiltered Search**: Discovery of all registered data sets, irrespective of Cube domain.

3. **Filtered Search**: Discovery of all registered data sets for a particular sub-domain.

5.4.1 Viewing an Existing Taxonomy

The 4-D Wx Data Cube's domains can be organized in and viewed as a taxonomy. This test performs the viewing of a pre-loaded taxonomy using the Registry Admin UI.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

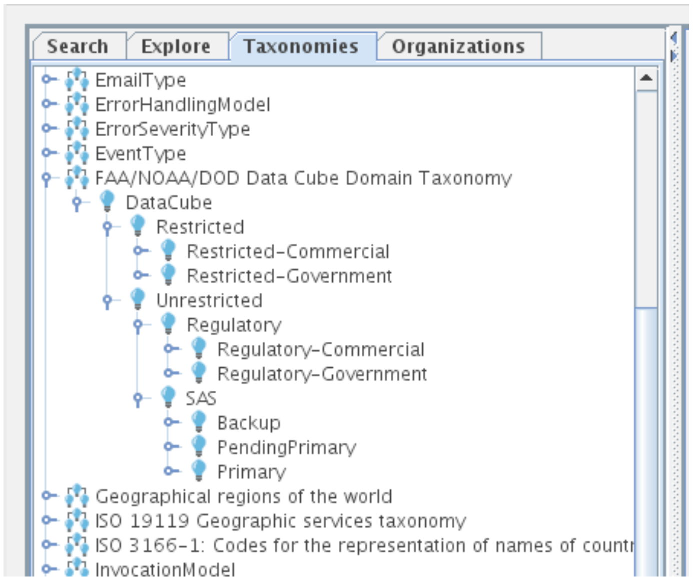
2. Click on the **Taxonomies** tab.

3. Scroll down to the **FAA/NOAA/DOD Data Cube Domain Taxonomy**.

4. Expand the **Taxonomy** tree node by clicking on its handle.

5. Verify that the structure of the taxonomy matches the test taxonomy that has been defined and stored in the Registry:

- DataCube
    - Restricted
        - Restricted-Commercial
        - Restricted-Government
    - Unrestricted
        - Regulatory
            - Regulatory-Commercial
            - Regulatory-Government
        - SAS
            - Backup
            - PendingPrimary
            - Primary



EmailType
ErrorHandlingModel
ErrorSeverityType
EventType
FAA/NOAA/DOD Data Cube Domain Taxonomy
DataCube
Restricted
Restricted-Commercial
Restricted-Government
Unrestricted
Regulatory
Regulatory-Commercial
Regulatory-Government
SAS
Backup
PendingPrimary
Primary
Geographical regions of the world
ISO 19119 Geographic services taxonomy
ISO 3166-1: Codes for the representation of names of countr
InvocationModel

Figure 5.7: Snapshot of the Registry Admin UI showing the test taxonomy

5.4.2 Dataset Discovery – Federated Search Filtered by Any Weather Cube Domain

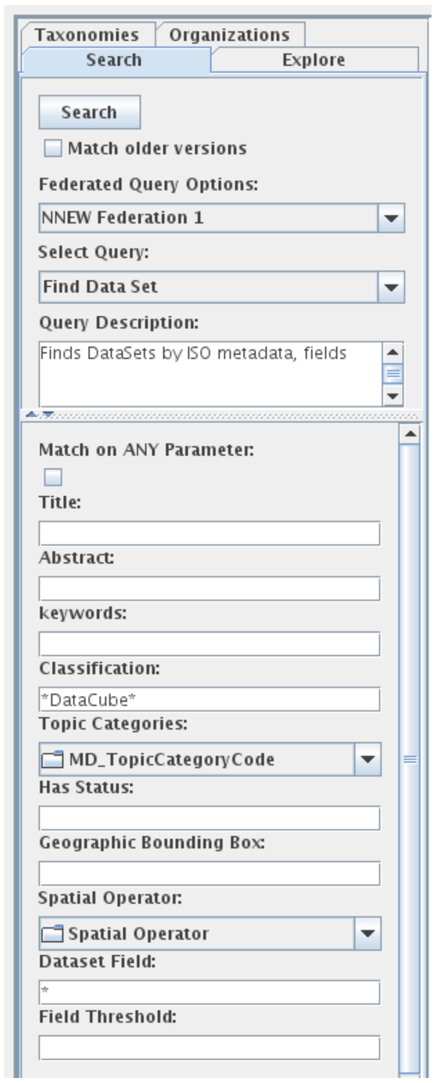
This test performs the discovery of all datasets registered as classified by the data cube domain taxonomy.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Click on the **Search** tab in upper-left corner of the UI to access the **Search Tool** panel.

3. Make sure that the **Federated Query Options** combo box shows **NNEW Federation 1** as its selection.

4. Select **Find Data Set** in the **Select Query** combo box.


Taxonomies
Organizations
Search
Explore

Search

Match older versions

Federated Query Options: NNEW Federation 1

Select Query: Find Data Set

Query Description: Finds DataSets by ISO metadata, fields

Match on ANY Parameter:

Title:

Abstract:

keywords:

Classification: *DataCube*

Topic Categories: MD_TopicCategoryCode

Has Status:

Geographic Bounding Box:

Spatial Operator: Spatial Operator

Dataset Field: *

Field Threshold:

Figure 5.8: Dataset Discovery – Federated Search Filtered by Any Cube Domain

5. Type the following text in the **Classification** field:

   ```
   *DataCube*
   ```

   Note: The '' is a wildcard character used to match zero or more arbitrary characters.*

6. Click the **Search** button. A filtered set of all registered datasets with weather phenomenon classified by the 4-D Wx Data Cube taxonomy are returned.

7. Click on the **Name** column heading to sort the datasets alphabetically by name.

8. All datasets are currently registered as members of the Cube. The following datasets are some of the results that should be returned:

   - Base Reflectivity 2km, Max Reflectivity
   - Base Reflectivity 2km, Optimal

- Base Reflectivity 4km, Max Reflectivity
- Base Reflectivity 4km, Optimal
- Composite Mosaic, 0-60k feet, 4km, Max Reflectivity
- Composite Mosaic, 0-60k feet, 4km, Optimal Mosaic
- Digital VIL Mosaic, 2km
- Digital VIL Mosaic, 4km

9. Write down the number of datasets that get returned:

____

5.4.3 Dataset Discovery – Federated Search Filtered by "Unrestricted" Cube Domain

This test performs the discovery of all datasets registered as members of the "Unrestricted" domain.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Click on the **Search** tab in upper-left corner of the UI to access the **Search Tool** panel.

3. Make sure that the **Federated Query Options** combo box shows **NNEW Federation 1** as its selection.

4. Select **Find Data Set** in the **Select Query** combo box.

5. Type the following text in the **Classification** field:

   ```
   *DataCube/Unrestricted*
   ```

6. Click the **Search** button.

7. All of the datasets that get returned are classified as being apart of the Unrestricted Data Cube. Write down the number of datasets that get returned:
   ____

8. Type the following text in the **Classification** field:

   ```
   *DataCube/Unrestricted/SAS*
   ```

9. Click the **Search button.**

10. All of the datasets that get returned may also contain the SAS classification so the number that

get returned may be the same. Write down the number of datasets that get returned:

____

11. Type the following text in the **Classification** field:

    `*DataCube/Unrestricted/SAS/Primary*`

12. Click the **Search** button.

13. Write down the number of datasets that get returned:

    ____

14. Clear out the **Classification** field.

15. Right-click on a **Find Data Set** tab and select **Close All**.

5.5 Discovery of Services Instances

A Web Coverage Service is any service capable of retrieving data as ISO "Coverages," which encompass gridded data, vertical wind profiles, measurements of weather phenomenon along a trajectory, and the like. The OGC Web Coverage Service and JMBL are two examples of services with these capabilities.
This test demonstrates discovery of weather services registered in the Registry/Repository. In particular, it demonstrates:

1. **Unfiltered Search**: Discovery of all registered service instances.

2. **Filtered Search**: Discovery of all registered service instances using the ISO 19119 taxonomy.

3. **Service Endpoint Retrieval**: Retrieval of the service endpoint for a specified service instance.

For brevity's sake, a "service instance" will be referred to as a "service" throughout the remainder of this document.

5.5.1 Service Discovery – Unfiltered Local Search

This test performs an unfiltered search for all registered services in the WJHTC Registry.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Click on the **Search** tab in upper-left corner of the UI to access the **Search Tool** panel.

3. Make sure that the **Federated Query Options** combo box shows **Local Query** as its selection.

4. Select **Find Service** in the **Select Query** combo box.

5. Click the **Search** button at the top of the **Search Tool** panel. An unfiltered list of all the registered services will appear. Click on the **Name** column heading to sort the datasets alphabetically by name.

6. Verify that the list of services returned includes the following:

   - FAA_TC_WebCoverageService
   - FAA_TC_WebFeatureService

5.5.2 Service Discovery – Unfiltered Federated Search

This test performs an unfiltered search for all registered services across all registries in the NNEW Federation 1.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Repeat the previous test Service Discovery – Unfiltered Local Search with one change: Make sure that in Step 3, the **Federated Query Options** combo box shows **NNEW Federation 1** as its selection.

3. Verify that the list of services returned includes services from the WJHTC, MIT-LL, NWS, and GSD registries, as indicated by the last column labeled **Registry** in the search results.

4. Right-click on the **Find Service** tab and select **Close**.

5.5.3 Service Discovery – Federated Search Filtered by Service Type

This test searches within the NNEW Federation for registered services, filtered by service type using the **ISO 19119 Geographic services taxonomy**.

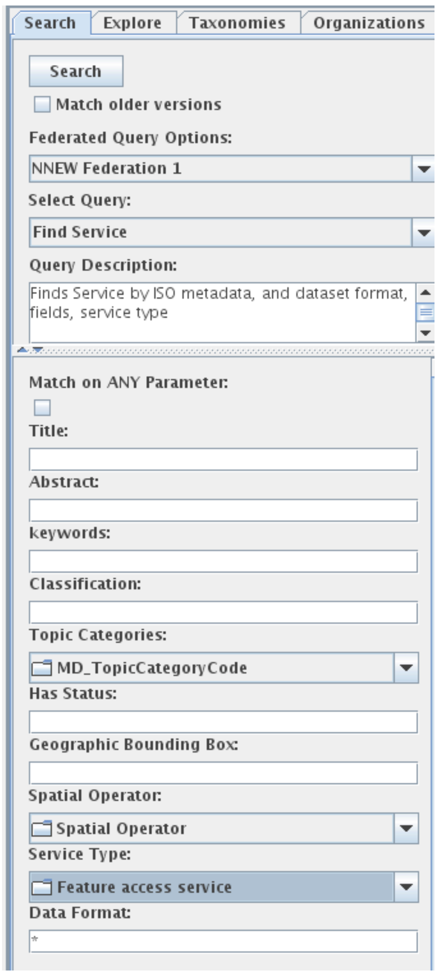
1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC registry.

2. Click on the **Search** tab in the upper-left corner of the UI to access the **Search Tool** panel.

3. Make sure that the **Federated Query Options** combo box shows **NNEW Federation 1** as its selection.

4. Select **Find Service** in the **Select Query** combo box.

5. Select **Feature Access Service** in the **Service Type** choice box.



Federated Query Options: NNEW Federation 1
Select Query: Find Service
Query Description: Finds Service by ISO metadata, and dataset format, fields, service type

Service Type: Feature access service
Data Format: *

Figure 5.9: Service Discovery – Federated Search Filtered by Service Type matches Feature Access Services

6. Click the **Search** button. A filtered set of all registered services with the service type **Feature Access Service** are returned.

7. Click on the **Name** column heading to sort the datasets alphabetically by name.

8. Verify that the result set includes the following services:

   2. DOD_JMBLService-01
   3. MIT_WebFeatureService-01
   4. NCAR_WebFeatureService-01

9. The Feature Access Service classification used in Step 5 is an abstract classification—it does not represent a particular implementation. To refine the query to return only OGC-compliant Feature Access Services (OGC WFS spec), specify the **Web Feature Service (WFS)** sub-type in the **Service Type** choice box.

10. Click the **Search** button.

11. Click on the **Name** column heading to sort the datasets alphabetically by name.

12. Verify that the results do not contain the DOD JMBL service as seen in the previous test.

13. Right-click on the **Find Service** tab and select **Close**.

14. Repeat Step 5, selecting **Coverage Access Services** in the **Service Type** choice box.

15. Click the **Search** button.

16. Click on the **Name** column heading to sort the datasets alphabetically by name.

17. Verify that the results include the following coverage services. Note that JMBL is capable of acting as both a feature and a coverage access service, so it appears in both coverage and feature access service requests.

    a) DOD_JMBLService-01
    b) MIT_WebCoverageService-01
    c) NCAR_WebCoverageService-01

18. The Coverage Access Service classification used in Step 14 is an abstract classification—it does not represent a particular implementation. To refine the query to return only OGC-compliant Coverage Access Services (OGC WCS spec), specify the **Web Coverage Service (WCS)** sub-type in the **Service Type** choice box.

19. Click the **Search** button.

20. Click on the **Name** column heading to sort the datasets alphabetically by name.

21. Verify that the results do not contain the DOD JMBL service as seen in the previous test.

22. Right click on the Find Service tab and select **Close**.

5.5.4 Service Endpoint Retrieval

This test retrieves the service endpoint for a selected service.

1. Query for services using the test Service Discovery – Unfiltered Local Search, as outlined in Section 5.5.1 or look through the past search results for FAA_TC_WebCoverageService.

2. Left-click on the FAA_TC_WebCoverageService service endpoint in the result set pane to select it. Right-click and select **Open** to open a more detailed service end-point sub-pane.

3. View the information presented in the top-right information pane.

4. Left-click on the entry in **Endpoints** to select it.

5. Verify that for FAA_TC_WebCoverageService the **endpoint** URL is:

   http://wcs.wjhtc.nnew.faa.gov:8280/wcs/soap

5.5.5 Viewing Datasets Related to a Service

This test verifies that a user can view datasets related to a service. It also illustrates how to view any object that is related to an object using an Association (relationship), as defined by ebRIM in ebXML RegRep.

1. Query for services using the test Service Discovery – Unfiltered Local Search, as outlined in Section 5.5.1 or look through the past search results for FAA_TC_WebCoverageService.

2. Left-click on the FAA_TC_WebCoverageService service in the result set pane to select it. Right-click and select **Open** to open a more detailed service end-point sub-pane.

3. View the information presented in the top-right information pane.

4. Click on the **Detail** button to show additional details about the service.

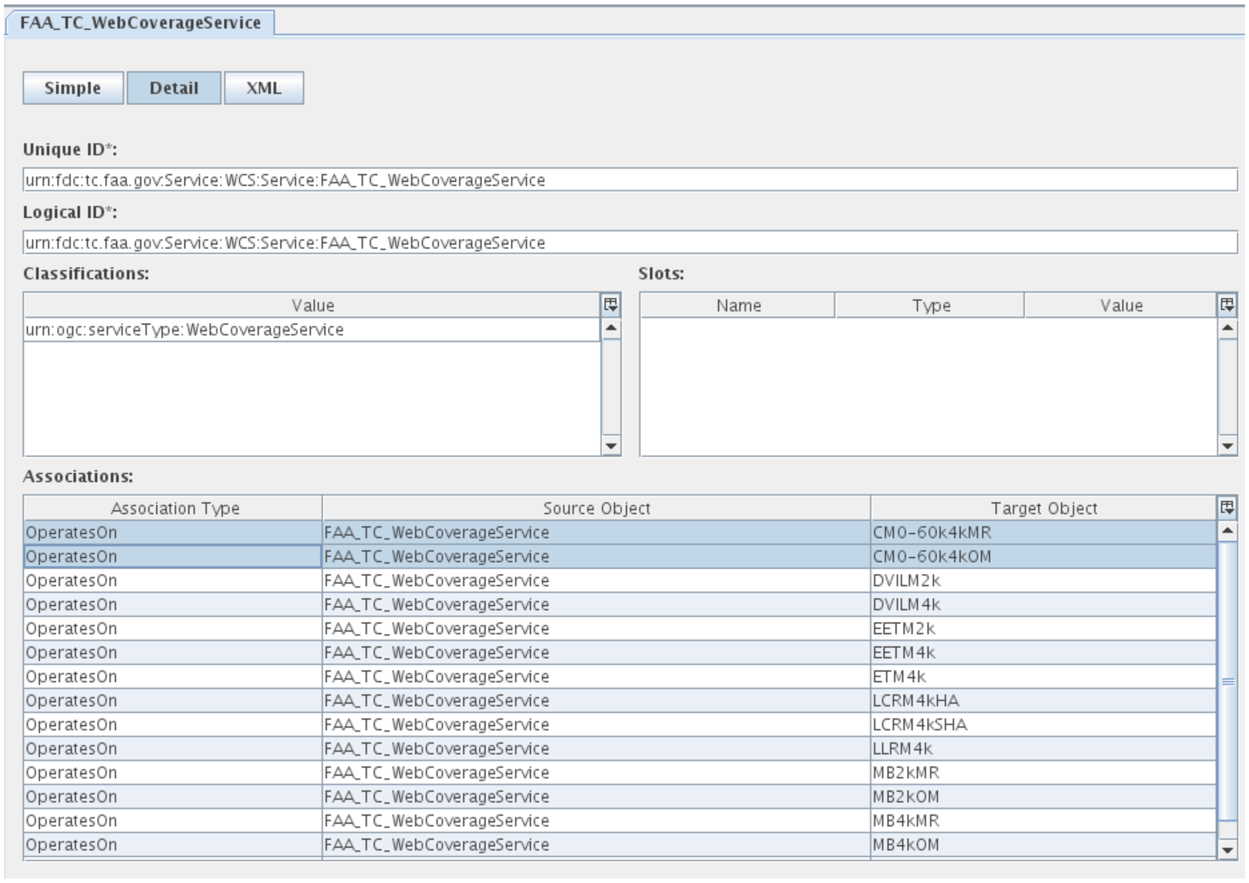
5. Verify that the Associations table shows associations of type **OperatesOn** with Target Objects that include:

   - CMO-60k4kMR
   - CMO-60k4kOM
   - DVILM2k
   - DVILM4k
   - EETM2k
   - EETM4k
   - ETM4k
   - LCRM4kHA

- LCRM4kSHA
- LLRM4k
- MB2kMR
- MB2kOM
- MB4kMR
- MB4kOM

6. Press and hold the Control key (to select multiple rows), then left click the CMO-60k4kMR and CMO-60k4kOM associations in the table to select them. Right-click one of the highlighted associations to display the context menu, and select the **Open Related Objects** menu from the available options.

7. Verify that the CMO-60k4kMR and CMO-60k4kOM datasets are displayed in new tabs within the editor panel.



Unique ID*:
urn:fdc:tc.faa.gov:Service:WCS:Service:FAA_TC_WebCoverageService

Logical ID*:
urn:fdc:tc.faa.gov:Service:WCS:Service:FAA_TC_WebCoverageService

Classifications:

Value
urn:ogc:serviceType:WebCoverageService

Slots:

Name	Type	Value

Associations:

Association Type	Source Object	Target Object
OperatesOn	FAA_TC_WebCoverageService	CMO-60k4kMR
OperatesOn	FAA_TC_WebCoverageService	CMO-60k4kOM
OperatesOn	FAA_TC_WebCoverageService	DVILM2k
OperatesOn	FAA_TC_WebCoverageService	DVILM4k
OperatesOn	FAA_TC_WebCoverageService	EETM2k
OperatesOn	FAA_TC_WebCoverageService	EETM4k
OperatesOn	FAA_TC_WebCoverageService	ETM4k
OperatesOn	FAA_TC_WebCoverageService	LCRM4kHA
OperatesOn	FAA_TC_WebCoverageService	LCRM4kSHA
OperatesOn	FAA_TC_WebCoverageService	LLRM4k
OperatesOn	FAA_TC_WebCoverageService	MB2kMR
OperatesOn	FAA_TC_WebCoverageService	MB2kOM
OperatesOn	FAA_TC_WebCoverageService	MB4kMR
OperatesOn	FAA_TC_WebCoverageService	MB4kOM

Figure 5.10: Viewing Datasets Related to a Service

5.6 Creation of an Experimental Weather Cube Taxonomy

The research community is continually refining old weather products and generating new ones. It is often convenient to have a test version of the 4D Wx Data Cube available that can provide information about experimental data sets. The NNEW Registry is capable of storing information about an arbitrary number of weather domains, in the form of taxonomies.

An experimental taxonomy is provided with the test data package to test the loading of custom weather domains. As constructed, it essentially mimics the official Cube domain, using different unique identifiers for each taxonomy node. Note that this is for test purposes only—custom weather domains are not necessarily limited to the same structure as the official weather domain taxonomy.

This test demonstrates the loading of a new taxonomy and verifies that the loaded taxonomy is similar to the Cube domain taxonomy:

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Login to the registry using a valid User ID and Password as a registry user.

3. Select **Tools → Wizards → Import ebRIM File** to open the file import window.

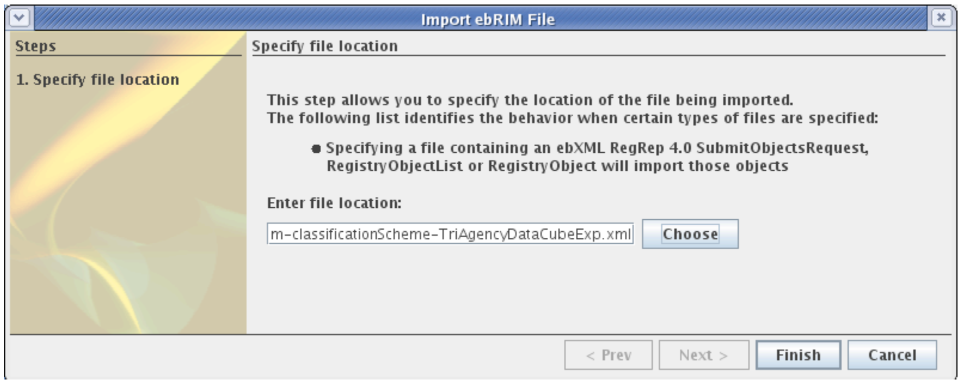
4. Browse to the `$TEST_DATA_DIR/wxcube-metadata/trunk/src/main/resources/taxonomies` directory provided with the test package.

5. Select the experimental version of the data cube domain taxonomy:

   ebrim-classificationScheme-TriAgencyDataCubeExp.xml

6. Click on **Open** and then **Finish** to perform the load. After a short pause, an indication of load success is returned.

7. Close the file import window.



This step allows you to specify the location of the file being imported.
The following list identifies the behavior when certain types of files are specified:

- Specifying a file containing an ebXML RegRep 4.0 SubmitObjectsRequest, RegistryObjectList or RegistryObject will import those objects

m-classificationScheme-TriAgencyDataCubeExp.xml

Figure 5.11: Snapshot of the Registry Admin UI displaying the interface for loading taxonomies

8. While the taxonomy now resides in the registry, the Admin UI session is not automatically aware of it. If the UI is restarted, it will be loaded; in order to refresh the Registry Admin UI

without restarting the application, use **View → Reload**. This operation takes approximately 30 seconds.

9. Click the **Taxonomies** tab. Navigate to **NOAA/NCAR/LL Experimental Data Cube Taxonomy**.

10. Browse the taxonomy tree and verify that it matches up with the official weather cube taxonomy:

- DataCube
    - Restricted
        - Restricted-Commercial
        - Restricted-Government
    - Unrestricted
        - Regulatory
            - Regulatory-Commercial
            - Regulatory-Government
        - SAS
            - Backup
            - PendingPrimary
            - Primary

5.7 Publication of an Experimental Data Set and Accompanying Experimental Data Access Service

This test demonstrates the publication of an experimental data set and an accompanying data access service. In order to demonstrate publication of datasets and related services, metadata for an additional (not pre-loaded) experimental dataset and service is included in the test package.

Note that although this test exercises dataset and service publication in an experimental context, the overall actions and results are the same for non-experimental datasets.

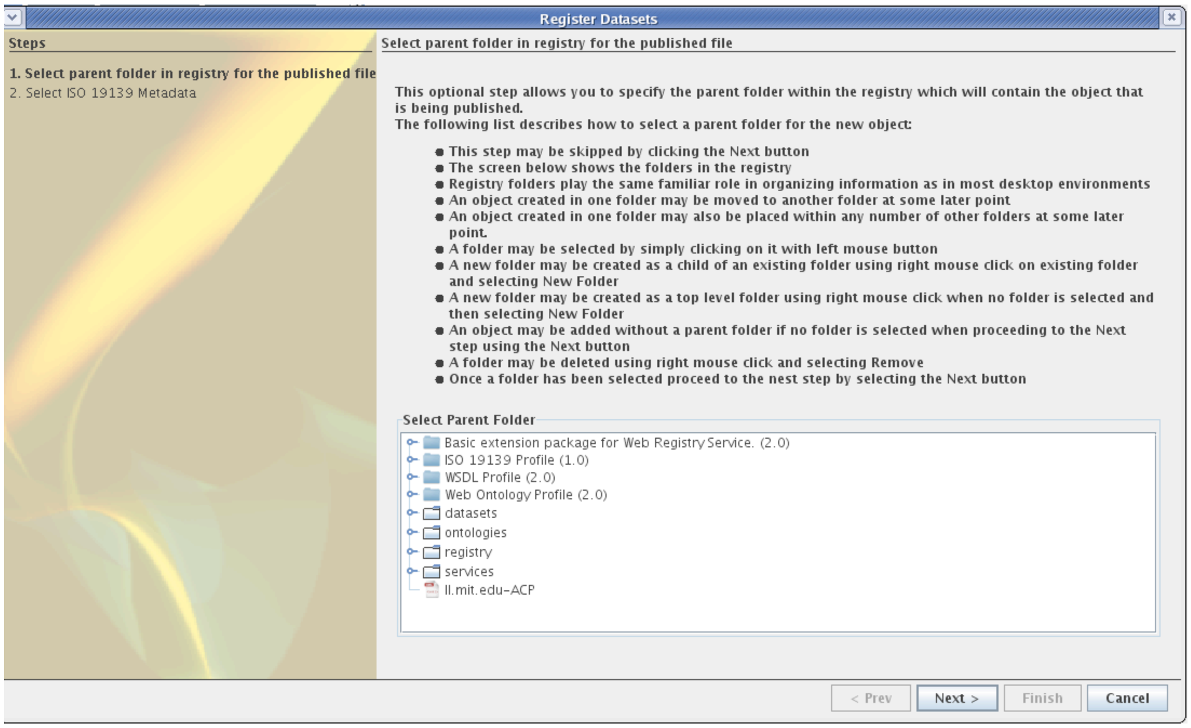
5.7.1 Publish Dataset

This test demonstrates the publication of an experimental data set.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Login with a valid **User ID** and **Password** using the login controls on the upper-right corner of the screen.

3. Open the **Register Datasets** wizard, clicking **Tools → Wizards → Register Datasets** in the menu bar.


Register Datasets
Steps
1. Select parent folder in registry for the published file
2. Select ISO 19139 Metadata
Select parent folder in registry for the published file

This optional step allows you to specify the parent folder within the registry which will contain the object that is being published.
The following list describes how to select a parent folder for the new object:

- This step may be skipped by clicking the Next button
- The screen below shows the folders in the registry
- Registry folders play the same familiar role in organizing information as in most desktop environments
- An object created in one folder may be moved to another folder at some later point
- An object created in one folder may also be placed within any number of other folders at some later point.
- A folder may be selected by simply clicking on it with left mouse button
- A new folder may be created as a child of an existing folder using right mouse click on existing folder and selecting New Folder
- A new folder may be created as a top level folder using right mouse click when no folder is selected and then selecting New Folder
- An object may be added without a parent folder if no folder is selected when proceeding to the Next step using the Next button
- A folder may be deleted using right mouse click and selecting Remove
- Once a folder has been selected proceed to the nest step by selecting the Next button

Figure 5.12: Dataset Publish – Select Parent Folder

4. Click on the **Next** button to skip the wizard's optional Step 1 (**Select Parent Folder**) and move on to its Step 2 (**Select ISO 19139 Metadata**), as shown in the **Steps** window on left-hand side of wizard.

5. Browse to `$TEST_DATA_DIR/wxcube-metadata/trunk/src/main/resources/datasets/mit-ll` and select the file:

   iso-metadata-dataset-EchoTops-Exp-01.xml

6. Click on the **Finish** button and then **Close** to publish the dataset specified in the ISO 19139 metadata file. An indication of success is returned on the Summary page.

7. Verify that the dataset was successfully published by searching for all datasets in the NNEW Registry by running the test from Section 5.3.1 . In addition to the previously available datasets, a new one should be present, entitled **Echo Tops (Experimental)**.

5.7.2 Publish Service Instance

This test publishes a service instance to the registry. Publication of services currently requires that two items be specified: a WSDL file and an accompanying ISO 19139 metadata file. In the **Publish Service Instance** wizard, the WSDL file is specified first, followed by the ISO 19139 file.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will

connect with the WJHTC Registry.

2. Login with a valid **User ID** and **Password** using the login controls in the upper-right corner of the screen.

3. Open the **Register Service Instance** wizard by clicking **Tools → Wizards → Register Service Instance** in the menu bar.

4. Click on the **Next** button to skip the wizard's optional Step 1 (**Select Parent Folder**) and go to its Step 2 (**Select Service Instance WSDL file**).

5. Specify the WSDL file for the service instance. Browse to `$TEST_DATA_DIR/wxcube-metadata/trunk/src/main/resources/services/mit-ll` and select the appropriate WSDL file: **MIT-WCS-03.wsdl**.

6. Select **Next** to go to **Select ISO 19139 Metadata** step.

7. Specify the ISO metadata file. Browse to `$TEST_DATA_DIR/wxcube-metadata/trunk/src/main/resources/services/mit-ll`, (this should still be there by default from the previous step) and select the appropriate ISO 19139 file: **iso-metadata-service-MIT-WCS-03.xml**.

8. Click on the **Finish** button and then **Close** to publish the service instance specified by the WSDL and the ISO 19139 metadata file. An indication of success is returned on the Summary page.

9. Verify that the service was successfully published by searching for all services in the NNEW Registry using the test from Section 5.5.1 . In addition to the previously available services, a new one should be present, entitled **MIT_WebCoverageService-03**, with a description that includes an indication that it is the new experimental service.

5.8 Fault Tolerance Support in Registry Client API

This section verifies that the fault tolerance capabilities of the registry client API—available from the wxForge wxconsumer project—support fault-tolerant NNEW Registry access. It assumes that the wxconsumer project has been checked out of the wxForge SVN repository, with its root directory at `$WXCONSUMER`.

The tests in this section use a command line Java program called **RegistryTestClient**, which performs a set of queries against its target registry. The program uses the regrep Java client library developed by MIT Lincoln Labs. This library provides the RegistryTestClient the ability to specify an infinite chain of backup registries, in addition to a target registry. It first attempts to send requests to the target registry—the first in the backup chain—but re-routes them sequentially through the chain if the target registry is unavailable or if connections to it time out after a configurable period of time. The re-routing process repeats until a registry in the backup chain is able to respond to the request within the configured timeout period. This test assumes that the computer running the RegistryTestClient program has Maven installed.

A client program specifies the backup registries and timeout period via a configuration file. In the case of the **RegistryTestClient**, this configuration file is specified via the "-c" option. For the following test, we use a configuration file that specifies:

- A backup chain consisting of the following sequence of registries:

    1. **WJHTC-Primary** – The primary WJHTC registry. It is configured with an incorrect URL with a nonexistent host to simulate a scenario in which the WJHTC Registry is unavailable.

    2. **WJHTC-Backup1** – The backup WJHTC Registry.

- 20 seconds as the timeout period for all registries in the backup chain.

1. If the regrep4 client library has been installed, skip to Step 2. Otherwise, open a terminal window and change the current working directory to `$WXCONSUMER/trunk/`. Run the command "`mvn install`".

2. Change the current working directory to `$WXCONSUMER/trunk/regrep/client`.

3. Type the following command to run the **RegistryTestClient** test with the specified configuration file:

```
./RegistryTestClient -c src/test/resources/registry-config-
faultTolerant.xml | tee /tmp/test.log
```

Note that the configuration file specifies that client requests are to be sent to the WJHTC Registry. Should it be unavailable, they are to be rerouted to the backup registry. The specified URL for the WJHTC Registry is deliberately incorrect, simulating that the WJHTC Registry is unavailable. The script file also assumes that the Maven repository on the current system is in its default location in the home directory under `.m2/repository`. If this is not the case, this file must be edited to point to the repository location.

The output of the command is logged in the `/tmp/test.log` file.

4. Read the `/tmp/test.log` file and verify that the following registries have been blacklisted due to being unreachable:

```
Blacklisting registry 'FAA-Primary'
…
```

5. Jump to the end of the `/tmp/test.log` file and verify that no errors are reported and the final output looks like this:

```
/DataCube/Unrestricted/Regulatory/Commercial
Registry not writeable - done
```

5.9 Default Access Control Policy

All **RegistryObjects** that do not define a custom access control policy are managed by a default policy. This policy defines the following:

- All read operations are allowed.
- Only logged-in users can create new content.
- Only the owners of a **RegistryObject** and administrators of the Registry can update or delete content.

The following tests demonstrate the default policy through the read, create, update, and delete operations, from the perspectives of different users.

5.9.1 Read Access

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. In the **Explore** tab, browse through the **RegistryObjects**. All of the **RegistryObjects** that do not have custom access control policies should be visible.

5.9.2 Create Access

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Open the **Import Directory Tree** wizard using **Tools → Wizards → Import Directory Tree** in the menu bar.

3. An error box should appear with the message “You must login before performing this action.”

4. Login with a valid **User ID** and **Password** using the login controls in the upper-right corner of the screen (user “rod”).

5. On the local file system, create a folder named `test` containing an empty text file: `test.txt`.

6. Open the **Import Directory Tree** wizard by clicking **Tools → Wizards → Import Directory Tree** in the menu bar.

7. In the **Import Directory Tree** wizard, click the **Choose** button and navigate to the `test` folder on the local file system created in Step 5. Click on **Open**, then the **Next** button.

8. In the window of files listed in the directory, check only the box next to the file `test.txt`. Click the **Finish** button. A success message should be displayed. Click the **Close** button.

9. In the **Explore** tab of the Admin UI, the folder that contains the file `test.txt`, along with the file itself, should be present.

5.9.3 Update Access

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Login with a valid **User ID** and **Password** using the login controls in the upper-right corner of the screen. Do not use the same user as in Section 5.9.2 (use the user “dianne”).

3. In the **Explore** tab, select the **RegistryObject** for `test.txt` created in test 5.9.2 , right-click it, and select **Open**. In the editor pane, change the description for `test.txt`. Right-click the editor pane and select **Save** from the menu. An error box should appear with the message “Error during publish. Are you logged in?” Click the **Close** button in the error box.

4. Login with the same **User ID** and **Password** from Section 5.9.2 (user “rod”).

5. Repeat Step 3. This time, the update should succeed.

5.9.4 Delete Access

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Login with a valid **User ID** and **Password** using the login controls in the upper-right corner of the screen. Do not use the same user as in Section 5.9.2 (use the user “dianne”).

3. In the **Explore** tab, right-click `test.txt` and select **Remove** from the menu. Click the **Yes** button in the confirmation box. An error box should appear with the message “Error during remove. Are you logged in? Are you authorized to remove this object? Check server logs for details.” Click the **Close** button in the error box.

4. Login with the same **User ID** and **Password** from Section 5.9.2 (user “rod”).

5. Repeat Step 3. A dialog box should appear to confirm the delete. Click the **Yes** button. This time, the **RegistryObject** for `test.txt` should be removed.

5.9.5 Administrator Access

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Login with a valid **Administrator ID** and **Password**.

3. In the **Explore** tab, select the **RegistryObject** for the directory named `test` created in Section 5.9.2 . Right-click the **RegistryObject** and select **Open**. In the editor pane, change the description for `test`. Right-click the editor pane and select **Save** from the menu. The update should occur with no errors.

4. Right-click the **RegistryObject** for `test` and select **Remove** from the menu. A dialog box should appear to confirm the delete. Click the **Yes** button. The **RegistryObject** for `test` should be removed.

5.10 Custom Access Control

The NNEW Registry allows for the creation of custom access control policies. These policies can be assigned to **RegistryObjects** to override the default access control policy. The following access control tests demonstrate specifically:

1. **Commercial Access Restriction –** To prevent users from accessing service endpoints for services that they have not paid for.

2. **Classification Access Restriction –** To prevent users from accessing information that is marked as being at a classification level above that of the user.

3. **Organizational Boundary Restriction –** To prevent users from one organization from manipulating data owned by another.

This test assumes that the LDAP instance supporting authentication for the Registry has the correct users and groups defined.

5.10.1 Commercial Test

Some of the WSDL and metadata documents in the Registry may describe connection information for a service that requires a paid subscription to access. This test demonstrates that a service that is marked as having a restricted legal constraint cannot be read by users who do not have a subscription. To allow this, an LDAP group was created that identifies users who have a subscription to Lightning. Note that this test assumes that the ISO 19139 metadata file was properly formatted to include legal constraints.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.1.5. By default, it will connect with the WJHTC Registry.

2. Select the **Tools → Options** menu item. In the **Registry Base URL** combo box, set the URL to the MIT-LL Registry.

3. Click on the **Search** tab in upper-left corner of the UI to access the Search Tool panel.

4. Make sure that the **Federated Query Options** combo box shows **Local Query** as its selection.

5. Select **Get RegistryObject By ID** in the **Select Query** combo box.

6. Set the **ID** field to `urn:fdc:ll.mit.edu:Service:WFS-Test:Service:MIT_WebFeatureService-Test`.

7. Click the **Search** button. No results should be returned.

8. Login with a valid **User ID** and **Password** for a user that is *not* a member of the subscription group (user "dianne").

9. Repeat Steps 4 through 6. No results should be returned.

10. Login with a valid **User ID** and **Password** for a user that *is* a member of the subscription group (user "rod").

11. Repeat Steps 4 through 6. The search results should return **MIT_WebFeatureService-Test**.

5.10.2 Classification Test

Some of the content contained in the NNEW Registry might have classification levels as security constraints. **RegistryObjects** can be marked as being *Restricted*, *Confidential*, *Secret*, or *Top Secret*. A user who does not have an equivalent security clearance or higher cannot perform actions on classified **RegistryObjects**.

This test assumes that LDAP groups were created for the different classification levels. It also assumes that the test ISO 19139 metadata documents are properly configured with the correct classification levels.

1. Launch the Registry Admin UI **version 4.7** as described in Section. By default, it will connect with the WJHTC Registry.

2. Select the **Tools → Options** menu item. In the **Registry Base URL** combo box, set the URL to the MIT-LL Registry.

3. Click on the **Search** tab in upper-left corner of the UI to access the **Search Tool** panel.

4. Make sure that the **Federated Query Options** combo box shows **Local Query** as its selection.

5. Select **Find ISO 19139 Metadata Document** in the **Select Query** combo box.

6. In the Title field, enter “`*Product*`”.

7. Click the **Search** button at the top of the **Search Tool** panel. An unfiltered list of all the registered metadata objects that have “Product” in their name will appear. Click on the **Name** column heading to sort the datasets alphabetically by name.

8. Confirm that the only result was:

   - Product

9. Login with a valid **User ID** and **Password** for a user that is a member of the *Restricted* classification group (user “restricted”).

10. Repeat Steps 5 through 7.

11. Verify that the following metadata documents are returned:

    - Product
    - Product R

12. Login with a valid **User ID** and **Password** for a user that is a member of the *Confidential* classification group (user “confidential”).

13. Repeat steps 5 through 7.

14. Verify that the following metadata documents are returned:

    - Product
    - Product C
    - Product R

15. Login with a valid **User ID** and **Password** for a user that is a member of the *Secret* classification group (user “secret”).

16. Repeat steps 5 through 7.

17. Verify that the following metadata documents are returned:

    - Product
    - Product C
    - Product R

- Product S

18. Login with a valid **User ID** and **Password** for a user that is a member of the *Top Secret* classification group (user “topsecret”).

19. Repeat steps 5 through 7.

20. Verify that the following metadata documents are returned:

    - Product
    - Product C
    - Product R
    - Product S
    - Product TS

21. Clear out the Title field.

5.10.3 Organizational Boundary Test

The access control policy can prevent members of other organizations from creating, updating, or deleting **RegistryObjects** from or for another organization.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.1.5. By default, it will connect with the WJHTC Registry.

2. Select the **Tools → Options** menu item. In the **Registry Base URL** combo box, set the URL to the MIT-LL Registry.

3. Login with a valid **User ID** and **Password** for a user that is a member of NCAR (user “ncarPerson1”).

4. Click on the **Search** tab in upper-left corner of the UI to access the **Search Tool** panel.

5. Make sure that the **Federated Query Options** combo box shows **Local Query** as its selection.

6. Select the query **Get Registry Object By ID** in the **Select Query** combo box. Enter `urn:fdc:ll.mit.edu:Dataset:Product-0015170B74D4` in the **ID** field. Click **Search.**

7. In the returned results, left-click the dataset to select it, then right-click and select **Open**.

8. In the editor pane, change the description. Right-click in the editor pane and select **Save**. An error box should appear with the message, “Error during publish. Are you logged in?” Click the **Close** button in the error box.

9. Right-click the dataset in the search results and select **Remove**. When prompted whether you wish to permanently remove the object, click **Yes**. An error box should appear with the message, “Error during remove. Are you logged in? Are you authorized to remove this object? Check stack trace and server logs for details.” Click the **Close** button.

10. Log out and then log in with a valid **User ID** and **Password** for a user that is a member of MIT but that is different from the user that uploaded the documents (user “mitPerson2”).

11. Repeat Step 8. This time the save should not cause an error box to appear.

5.11 Schematron-Based XML Validator Plugin

The NNEW Registry supports validation of XML content that is submitted to it. XML documents like ISO 19139 metadata files, WSDLs, and other files are validated and any errors are reported back to the user. This test checks that non-valid content submitted to the Registry causes errors to be thrown and the transaction to fail.

1. Launch the Registry Admin UI **version 4.7** as described in Section 5.2.5 . By default, it will connect with the WJHTC Registry.

2. Login with a valid **User ID** and **Password** (user “rod”).

3. Browse to `$TEST_DATA_DIR/trunk/src/main/resources/datasets/test`, and select the file:

   iso-metadata-dataset-validation-test.xml

4. Open this file in a text editor and remove one of the angle brackets (<) from the XML. Remember which bracket you remove. Save the file.
5. Open the **Register Datasets** wizard by clicking **Tools → Wizards → Register Datasets** in the menu bar.

6. Click on the **Next** button to skip the wizard’s optional Step 1 (**Select Parent Folder**) and go on to Step 2 (**Select ISO 19139 Metadata**), as shown in the Steps window on the wizard’s left-hand side.
7. Click on the **Finish** button to publish the dataset specified in the ISO 19139 metadata file. An error box should appear with the message “Error during Publish.” Click the **Details** button and scroll through the stack trace. The trace should reveal the error in the document. Click on the **Close** button on the error dialog and then the **Close** button on the wizard.
8. Click on the **Search** tab in upper-left corner of the UI to access the **Search Tool** panel.

9. Make sure that the **Federated Query Options** combo box shows **Local Query** as its selection.

10. Select the query **Get Registry Object By ID** in the **Select Query** combo box. Enter `urn:fdc:ll.mit.edu:Dataset:validation-test:metadata` in the **ID** field.

Click **Search.** No results should be returned.

11. Replace the angle bracket you removed from the file iso-metadata-dataset-validation-test.xml. Save the file and rerun steps 5 through 9. This time the file should publish successfully and the search should returned the validation test dataset.

6 SA Test Procedures for RASP, DOTS and ATOP

6.1 Introduction

6.1.1 Purpose and Scope

The purpose of these procedures is to demonstrate that the RASP, DOTS, and ATOP Service Adapters can meet the accuracy and throughput performance objectives. Specific tests will be developed for each SA to test these functional areas. Objectives will be associated with each specific test**.**

6.2 Reference Documents

N/A

6.3 Test Description

6.3.1 System Under Test

RASP:

The RASP SA encompasses two functions. The first is the Weather Data Gateway (WDG) Proxy and the second is the SA Processor. The WDG Proxy receives the data (Lightning Detection Data [LDD], Routine Aviation Weather Observation [METAR], and One Minute Observation [OMO]) from the RASP. The WDG Proxy then transmits this data to the SA Processor. The SA Processor prepares the LDD, OMO, and METAR data for publishing to the 4-D Wx Data Cube utilizing the Web Feature Service Reference Implementation (WFSRI).

DOTS:

The DOTS SA will subscribe to the WCSRI in order to pull GFS Thin Grids data in NetCDF4 file format. The DOTS SA will generate the appropriate GRIB1 files for U and V winds and temperature for the prescribed atmospheric levels. These files will be sent (FTP) to the DOTS servers for processing.

ATOP:

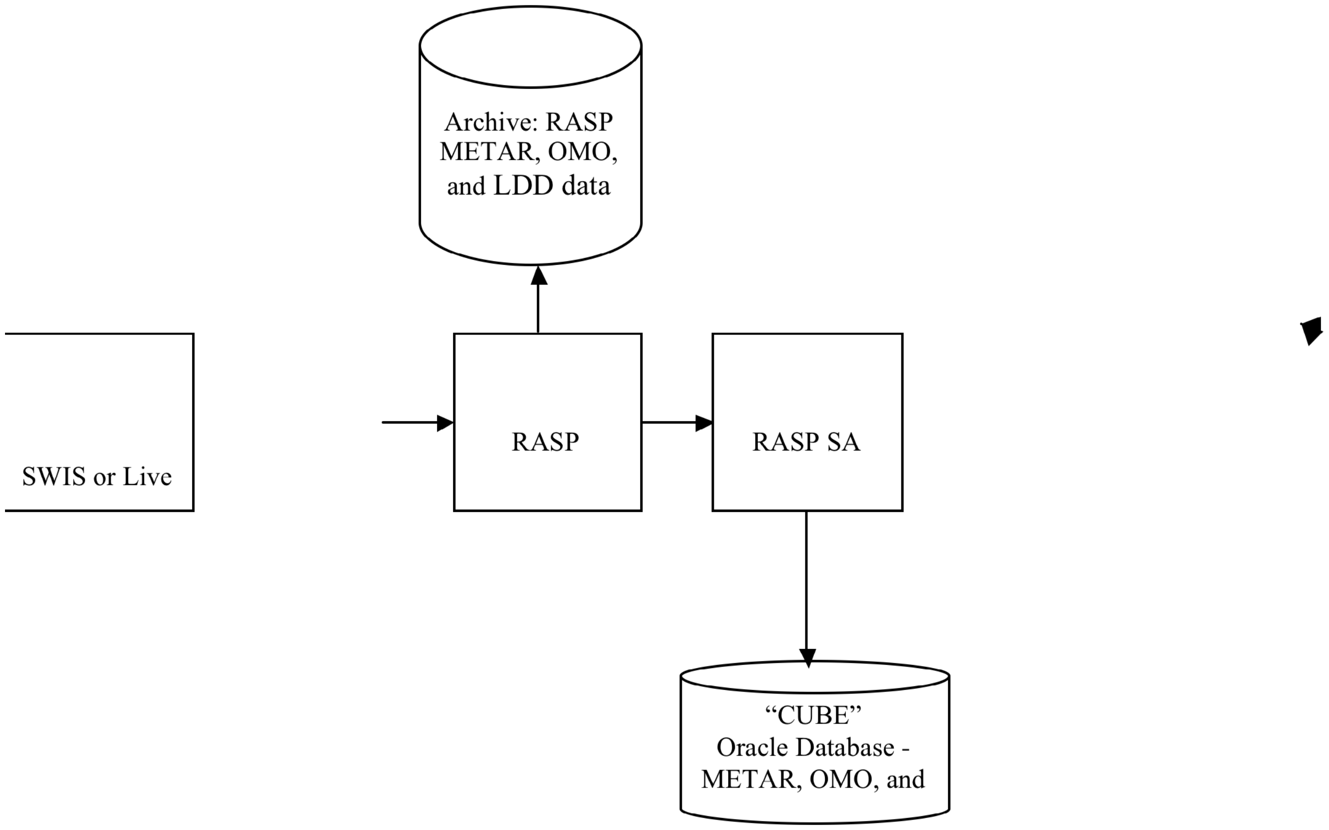
The ATOP SA will subscribe to the WCSRI in order to pull GFS and UKMET Thin Grids data in NetCDF4 file format. The ATOP SA will generate the appropriate GRIB1 files for U and V winds and temperature for the prescribed atmospheric levels. These files will be sent (FTP) to the ATOP servers for processing.

[Note: Since the DOTS and ATOP Services Adapters utilize the same adapter software, only one test will be run to verify these service adapters]

6.3.2 Test Setup

a. RASP

The RASP SA testbed will consist of the following:

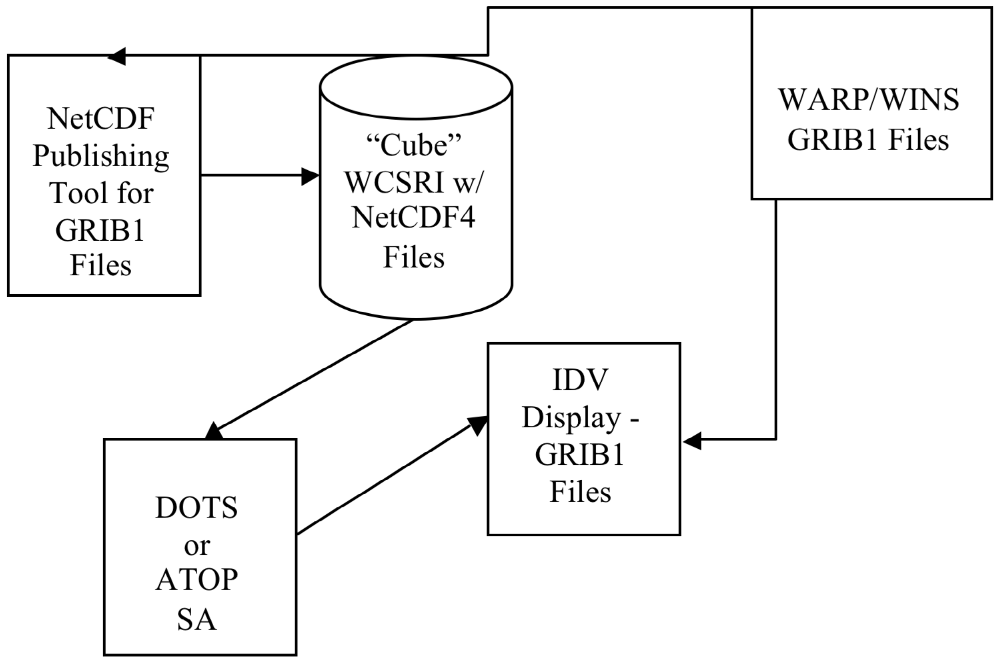

Archive: RASP METAR, OMO, and LDD data
SWIS or Live
RASP
RASP SA
"CUBE" Oracle Database - METAR, OMO, and

1. SWIS will be used to provide simulated LDD and OMO data to the RASP.

2. RASP will be used to process the LDD and OMO data. The RASP will process the OMO data and create the corresponding METARs. An Archive will be kept of the METAR/OMO and LDD data.
3. RASP SA:
    a. WDG Proxy will receive the LDD, OMO, and METAR data from the RASP and then transmit this data to the SA Processor.
    b. SA Processor will publish the LDD, OMO, and METAR data to the 4-D Wx Data Cube utilizing the WFSRI.

b. DOTS and ATOP

The DOTS and ATOP SA testbed will consist of the following:


NetCDF Publishing Tool for GRIB1 Files
"Cube" WCSRI w/ NetCDF4 Files
WARP/WINS GRIB1 Files
IDV Display - GRIB1 Files
DOTS or ATOP SA

1. NetCDF Publishing Tool will be used to publish GRIB1 files into the 4-D Wx Data Cube in the NetCDF4 file format. [Note: NWS sample NetCDF4 files will be used in place of the NetCDF Publishing Tool generated NetCDF4 files, if available].
2. DOTS and ATOP SA will be used to subscribe to appropriate NetCDF4 files and convert the data to the appropriate GRIB1 files.
3. IDV will be used to display GRIB1 files.

6.3.3 Test Equipment

N/A

6.3.4 Personnel

The tests will all be executed at the FAA Tech Center by Tech Center personnel. SA developers and test personnel will be used to verify that the objectives of these procedures have been met.

6.4 Test Conduct

6.4.1 Safety Considerations

N/A

6.4.2 Requirements Under Test

1. **METAR Accuracy Test, ME-01 and Live Metar and LDD Test, LV-01**

   OBJ-001: All RASP SA METAR messages published to the 4-D Wx Data Cube shall only contain fields that were part of the METAR messages that were sent by the RASP.

   OBJ-002: All RASP SA METAR messages published to the 4-D Wx Data Cube shall contain all fields that were part of the METAR messages that were sent by the RASP.

   OBJ-003: All RASP SA METAR messages published to the 4-D Wx Data Cube shall contain identical information for each field as the METAR messages that were sent by the RASP.

2. **LDD Accuracy Test, LD-01**

   OBJ-004: All RASP SA LDD messages published to the 4-D Wx Data Cube shall only contain fields that were part of the LDD messages that were sent by the RASP.

   OBJ-005: All RASP SA LDD messages published to the 4-D Wx Data Cube shall contain all fields that were part of the LDD messages that were sent by the RASP.

OBJ-006: All RASP SA LDD messages published to the 4-D Wx Data Cube shall contain identical information for each field as the LDD messages that were sent by the RASP.

3. DOTS Test, DT-01

OBJ-007: All GRIB1 files generated by the DOTS SA shall provide the same data as the original GRIB1 files that were published to the 4-D Wx Data Cube as NetCDF4 files.

OBJ-008: The DOTS SA shall properly respond to the new data notifications when NetCDF4 files are available in the 4-D Wx Data Cube.

OBJ-009: The DOTS SA shall publish the proper GRIB1 files to the DOTS at the predefined, 6 hour periods. [Note: This requirement is contingent on the necessary NetCDF4 files being available in the 4-D Wx Data Cube].

OBJ-010: The DOTS SA shall automatically begin subscribing to the 4-D Wx Data Cube once connected to the WCSRI.

OBJ-011: Once connected, the DOTS SA shall automatically begin pushing data to the DOTS server. [Note: contingent on OBJ-010 verified].

6.4.3 Procedures

a. METAR Accuracy Test, ME-01

The overall approach will include using a simulator to generate controlled, specific METAR messages and then comparing archived RASP messages to messages that the RASP SA pushes to the 4-D Wx Data Cube.

Details: The SWIS will be used to generate a simulation that contains 5 METAR messages/minute, one each for 5 different stations (K001 to K005). These 5 METAR messages will be varied in content for a 3 minute period. Each minute a new METAR message will be generated for each of these 5 stations. Once the simulation finishes, the archived RASP METAR messages will be compared to the METAR messages published to the 4-D Wx Data Cube by the RASP SA. This comparison will be used to determine if there are any anomalies and verify that the objectives have been met.

Test Procedures

1. P__ F__ Ensure the WFSRI and Oracle database are up and running.
2. P__ F__ Record the number of METAR messages in the database

   _____________.
3. P__ F__ Start the SWIS
4. P__ F__ Ensure the RASP is running
5. P__ F__ Ensure the RASP Proxy is configured for accepting data from the SWIS
6. P__ F__ Start the RASP Proxy
7. P__ F__ Start the RASP SA
8. P__ F__ Begin the ME-01 SWIS simulation
9. P__ F__ Record the date and time: date __________ time _____________
10. P__ F__ Wait 3 minutes and stop the simulation.

Data Analysis

1. P__ F__ Compare the archived RASP METAR messages (in the Proxy Log) with the METAR messages published to the 4-D Wx Data Cube by the RASP SA.
2. P__ F__ (OBJ-001) Verify, for every message, that no fields or extraneous information appears in the RASP SA METAR messages (published to the cube) that were not part of the original messages that came from the RASP.

3. P__ F__ (OBJ-002) Verify, for every message, that all fields appear in the RASP SA output METAR messages (published to the cube) that were part of the original messages that came from the RASP.

4. P__ F__ (OBJ-003) Verify, for every message, that all the data that appears in the RASP SA output METAR messages (published to the cube) are identical to the original messages that came from the RASP.

Test ME-01: P____ F____

Test Engineer: _____________________________ Date: ____________

b. LDD Accuracy Test, LD-01

The overall approach will include using a simulator to generate controlled, specific LDD messages and then comparing archived RASP messages to the messages that the RASP SA published to the 4-D Wx Data Cube.

Details: The SWIS will be used to generate a simulation that contains 5 LDD messages/minute, one each for 5 different locations. These 5 LDD messages will be varied in content for a 3 minute period. Each minute a new LDD message will be generated for each of these 5 stations. Once the simulation finishes, the SWIS LDD messages will be compared to the LDD messages published to the 4-D Wx Data Cube by the RASP SA. This comparison will be used to determine if there are any anomalies and verify that the objectives have been met.

Test Procedures

1. P__ F__ Ensure the WFSRI and Oracle database are up and running.

2. P__ F__ Clear the database of LDD messages to ensure no old data is stored.

3. P__ F__ Start the SWIS

4. P__ F__ Ensure the RASP is running

5. P__ F__ Ensure the RASP Proxy is configured for accepting data from the SWIS

6. P__ F__ Start the RASP Proxy

7. P__ F__ Start the RASP SA

8. P__ F__ Begin the LD-01 SWIS simulation

9. P__ F__ Record the date and time: date __________ time _____________

10. P__ F__ Wait 3 minutes and stop the simulation

Data Analysis

1. P__ F__ Compare the SWIS LDD messages (screen shots) with the LDD messages published to the 4-D Wx Data Cube by the RASP SA.
2. P__ F__ (OBJ-004) Verify, for every message, that no fields or extraneous information appears in the RASP SA LDD messages (published to the cube) that were not part of the original messages that came from the SWIS.
3. P__ F__ (OBJ-005) Verify, for every message, that all fields appear in the RASP SA output LDD messages (published to the cube) that were part of the original messages that came from the SWIS.
4. P__ F__ (OBJ-006) Verify, for every message, that all the data that appears in the RASP SA output LDD messages (published to the cube) are identical to the original messages that came from the SWIS.

Test LD-01: P____ F____

Test Engineer: ______________________________ Date: ____________

c. Live Throughput METAR and LDD Test, LV-01

The overall approach will include using live data coming from the RASP (WDG East and West) to provide the METAR/OMO and LDD messages. The RASP SA will be configured to receive the maximum number of messages available from the RASP. When the 10 minute live test is completed, the archived RASP METAR/OMO messages will be compared to the messages published to the 4-D Wx Data Cube to ensure that all live messages were published to the cube.

Test Procedures

1. P__ F__ Ensure the WFSRI and Oracle database are up and running.
2. P__ F__ Record the number of METAR messages in the database

   ____

3. P__ F__ Record the number of LDD messages in the database _____________
4. P__ F__ Ensure that the RASP Proxy is configured to provide live METAR/OMO and LDD data from both the WDG East and West.
5. P__ F__ Start the RASP Proxy
6. P__ F__ Start the RASP SA
7. P__ F__ Record the date and time: date __________ time _____________
8. P__ F__ Display a METAR message in XML form
9. P__ F__ Display latest LDD messages every few minutes
10. P__ F__ Wait 10 minutes before stopping data collection
11. P__ F__ Record the number of METAR messages in the database ____________
12. P__ F__ Record the number of LDD messages in the database _____________

Data Analysis

1. P__ F__ Select one station and compare 10 minutes of the archived RASP METAR messages (in the Proxy Log) with the corresponding METAR messages published to the 4-D Wx Data Cube by the RASP SA.

2. P__ F__ (OBJ-001) Verify, for every message that was checked, that no fields or extraneous information appears in the RASP SA METAR messages (published to the cube) that were not part of the original messages that came from the RASP.
3. P__ F__ (OBJ-002) Verify, for every message that was checked, that all fields appear in the RASP SA output METAR messages (published to the cube) that were part of the original messages that came from the RASP.
4. P__ F__ (OBJ-003) Verify, for every message that was checked, that all the data that appears in the RASP SA output METAR messages (published to the cube) are identical to the original messages that came from the RASP.

Test LV-01: P____ F____

Test Engineer: ______________________________ Date: ____________

d. DOTS (and ATOP) Test, DT-01

The overall approach will include using the NetCDF Publishing Tool to convert archived GFS Thin Grid GRIB1 files to NetCDF4 files [Note: For expediency, this will be done prior to running this test and only a subset of the 1320 GRIB1 files will be used]. This tool will publish the GFS Thin Grid data in NetCDF4 format to the 4-D Wx Data Cube [Note: NWS NetCDF4 files will not be available therefore, archived GRIB1 files will be converted using the NetCDF Publishing Tool]. The DOTS SA will subscribe to the 4-D Wx Data Cube to get these GFS Thin Grid NetCDF4 files. Once the DOTS SA receives notification, the SA will retrieve the NetCDF4 files with the temperature and U and V wind component data. The DOTS SA will then create the appropriate GRIB1 formatted files that DOTS requires. The DOTS SA will FTP (File Transfer Protocol) these GRIB1 files to the DOTS server. [NOTE: Since the DOTS system is not available for the CE, the DOTS SA will FTP the data to a simulated DOTS server].

To ensure that the DOTS SA generated GRIB1 files are the same as the original GRIB1 files, the IDV display tool will be used. The original GRIB1 files and the GRIB1 files that are generated by the DOTS SA will be displayed/overlaid on the IDV.

Comparison of these displayed files and directory listings of the files will be the means for verification of the requirements.

Test Procedures

1. P__ F__ Ensure the WCSRI is up and running.
2. P__ F__ Start the DOTS SA.
3. P__ F__ Start the subscription simulation client
4. P__ F__ Record the date and time: date __________ time _____________
5. P__ F__ Wait for the GRIB1 files to be processed (Wait for “Done Processing” message)

Data Analysis

1. P__ F__ Start the IDV
2. P__ F__ For temperature, use the IDV to display the original GRIB1 file for grid I, 150mb, and time 0.
3. P__ F__ For temperature, use the IDV to display the DOTS SA GRIB1 file for grid I, 150mb, and time 0.
4. P__ F__
5. (OBJ-007) Verify that the temperature displays are the same.
6. [NOTE: as a result of known translation/conversion issues going from GRIB1 to NetCDF and NetCDF back to GRIB1, the results will not exactly match, especially at the polar region]
7. P__ F__ For U component wind data, use the IDV to display the original GRIB1 file for grid I, 150mb, and time 0.

8. P__ F__ For U component wind data, use the IDV to display the DOTS SA GRIB1 file for grid I, 150mb, and time 0.
9. P__ F__ (OBJ-007) Verify that the U component wind data displays are the same.
10. [NOTE: as a result of known translation/conversion issues going from GRIB1 to NetCDF and NetCDF back to GRIB1, the results will not exactly match, especially at the polar region]
11. P__ F__ For V component wind data, use the IDV to display the original GRIB1 file for grid I, 150mb, and time 0.
12. P__ F__ For V component wind data, use the IDV to display the DOTS SA GRIB1 file for grid I, 150mb, and time 0.
13. P__ F__ (OBJ-007) Verify that the V component wind data displays are the same.
14. [NOTE: as a result of known translation/conversion issues going from GRIB1 to NetCDF and NetCDF back to GRIB1, the results will not exactly match, especially at the polar region]
15. P__ F__ (OBJ-008, OBJ-009, OBJ-010, and OBJ-011) Verify that all the original GRIB1 files are matched by corresponding GRIB1 files from the DOTS SA [Note: the filenames will be the same but the DOTS SA files will have a creation date/time from today]

Test DT-01: P____ F____

Test Engineer: ______________________________ Date: ____________

7 NEVS Procedures: Using NEVS as a Consumer

7.1 Introduction

7.1.1 Purpose and Scope

The purpose of this test is to demonstrate that the Networked Enabled Verification Service (NEVS) satisfies the capability evaluation requirements as specified in Appendix C of the Capability Evaluation Plan.

7.2 Reference Documents

- A ESRL/GSD Forecast Verification Section description can be found at http://esrl.noaa.gov/gsd/ab/fvs/.

- NEVS access can be found at: http://esrl.noaa.gov/nevs/demo/

7.3 Test Description

7.3.1 System Under Test

NEVS will be a consumer system which will be attempting to obtain the following:

Dataset: CCFP; Provider: NCAR; Service WFS; Public URL

Dataset: COSPA; Provider: MIT; Service WCS; Public URL

Dataset: LAMP; Provider: MDL; Service WCS; NOAAnet URL

Dataset: NCWD; Provider: AWC; Service WFS; NOAAnet URL

7.3.2 Test Setup

Prior to the test, the evaluation PC must be loaded with a supported web browser to access the NEVS web page via the internet.

a) Test Equipment at the Tech Center

- Evaluation PC running a web browser.

b) Test Equipment at remote site

- Service endpoints with live data

7.3.3 Personnel

FAA WJHTC, GSD

7.3.4 Client Location

NWEC lab at WJHTC

7.3.5 Server Location

NOAA/ESRL/GSD

7.4 Test Conduct

7.4.1 Security Concerns

None.

7.4.2 Requirements Under Test

The specified requirements for NEVS in Appendix C of the current FY10 Capability Evaluation Plan.

7.4.3 Procedures

1. Go to the NEVS landing page: http://esrl.noaa.gov/nevs/demo/
2. Load the NEVS synthesis demonstration page:
   http://esrl.noaa.gov/nevs/demo/synthesis/index.html
3. Confirm that 3 display panes showing mincut sector maps with observation fields are displayed and a single pane in the bottom right showing verification plots is displayed.
4. Click on the bottom right pane showing verification plots to enlarge the pane.
5. Check that CCFP and COSPA verification plots are meaningful.
6. Dismiss the verification plot graphic by clicking on the dismiss icon at the top

right.

7. Click on the “Status” tab at the bottom of the bottom right pane and confirm the plot changes to show data status plots.
8. Click on the bottom right pane showing data status plots to enlarge the pane.
9. Verify that WCS and WFS data ingest is being shown.
10. Dismiss the data status plot graphic by clicking on the dismiss icon at the top right.
11. Close browser tab/window and conclude test.

8 MDL Test Procedures

8.1 NDFD Wind Speed dataset

8.1.1 Introduction

8.1.1.1 Purpose and Scope

The purpose of this test is to demonstrate the way the 4-D Wx Data Cube may operate with National Digital Forecast Database (NDFD) Wind Speed data at initial operating capacity. The publishing and discovery of metadata about the WCS service and the NDFD Wind Speed dataset itself from federated reg/rep agency repositories will be tested. The requirements of the WCS RI are located in Appendix C of the Capability Evaluation Plan. System level requirements in Appendix D. Latencies, although not required, can be measured.

8.1.2 Reference Documents

1. A NWS NDFD program description can be found at http://www.nws.noaa.gov/ndfd/.
2. NDFD definitions can be found at http://www.nws.noaa.gov/ndfd/definitions.htm
3. NDFD data access can be found at: http://www.nws.noaa.gov/ndfd/technical.htm

8.1.3 Test Description

8.1.3.1 System Under Test

MDL will be the source system which will be providing NDFD/NDGD data via NOAAnet

8.1.3.2 Test Setup

Prior to the test, the evaluation PC must be loaded with RedHat 5 O/S and configured with OGC compliant software to request and display weather data from the Cube. Unidata's IDV must also be installed to verify that the dataset is available.

a. Test Equipment at the Tech Center

NAS Simulator, evaluation PC running the IDV, GSD's Testing Portal

b. Test Equipment at remote site

MDL's WCSRI endpoint with live data

8.1.3.3 Personnel

FAA WJHTC, MDL, GSD

8.1.3.4 Client Location

NWEC lab at WJHTC

8.1.3.5 Server Location

NOAA/NWS/MDL

8.1.4 Test Conduct

8.1.4.1 Security Concerns

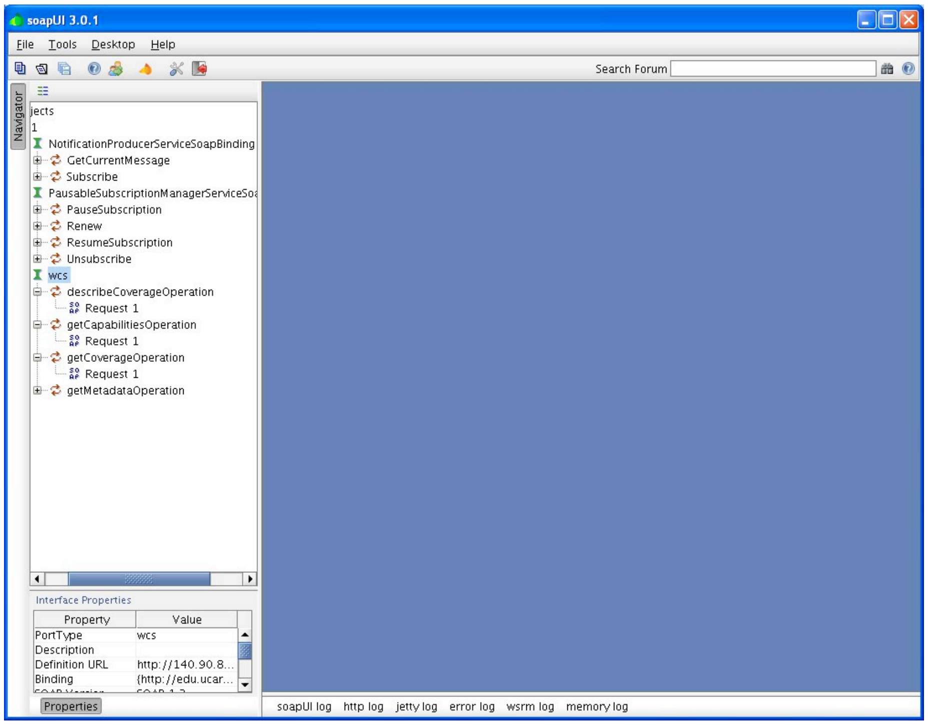
Performance on operational system – to get around this do it at 19Z. Least impact to FTI. Compliant with NIST guided, NOAA/FAA policies on security architecture and relevant information security technologies through a XML gateway.

8.1.4.2 Requirements Under Test

- RI shall support all WCSRI 1.1.0/2.0 spec requirements in Appendix C of the current FY10 Capability Evaluation Plan
- Though not required, the system performance (e.g. latency) shall be measured/baselined.

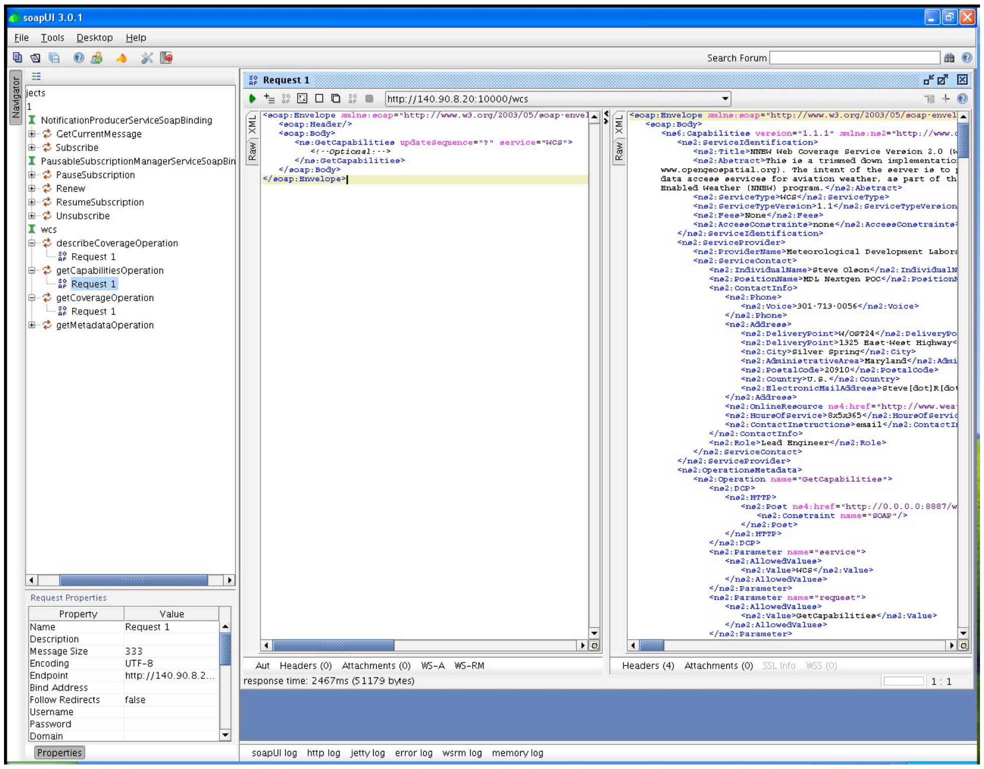
8.1.4.3 Procedures

1. Start up soapUI version 3.0+.
   - Import mdl-wcs-soapui-project.xml file if needed. Click on **File**, then **Import Project,** then select file **mdl-wcs-soapui-project.xml , and click on Open**.
2. Click on the "plus" sign and this will expand the **wcs** to include the following SOAP functions:
   - **describeCoverageOperation**
   - **getCapabilitiesOperation**
   - **getCoverageOperation**
   - **getMetadataOperation**



Property	Value
PortType	wcs
Description	
Definition URL	http://140.90.8...
Binding	{http://edu.ucar...

3. Click the "+" symbol for each of these features, and you will reveal a "SOAP Request 1". These soap requests serve as the interface between the user, and our WCS (where the data resides).
4. Step one: Perform a getCapabilities SOAP request using soapUI:
   - Double click on the "SOAP Request 1" associated with getCapabilitiesOperation. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server. There are no fields that need to be populated with this request. This is a simple request to find out what data is available and more background information about our WCS.
   - Click on the green arrow (Submit request to a specified endpoint URL). After execution, a getCapabilities soap response will appear in the right-hand pane indicating some background information about who's offering the service, and what data is available via our WCS.
   - The most important information that is contained in this soap response is the "ns2:parameter name="Identifier". This will provide "Values" indicating what data is available. The "urn: value form this soap response is an important input to the second request.
   - Copy the urn information (**`urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDFD-CONUS-Wind-Speed`**) into the copy and paste buffer.



http://140.90.8.20:10000/wcs

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envel
   <soap:Header/>
   <soap:Body>
      <ns:GetCapabilities updateSequence="?" service="WCS">
         <!--Optional:-->
      </ns:GetCapabilities>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envel
   <soap:Body>
      <ns6:Capabilities version="1.1.1" xmlns:ns2="http://www.c
         <ns2:ServiceIdentification>
            <ns2:Title>NNEW Web Coverage Service Version 2.0 (W
            <ns2:Abstract>This is a trimmed down implementatio
      www.opengeospatial.org). The intent of the server is to p
      data access services for aviation weather, as part of th
      Enabled Weather (NNEW) program.</ns2:Abstract>
            <ns2:ServiceType>WCS</ns2:ServiceType>
            <ns2:ServiceTypeVersion>1.1</ns2:ServiceTypeVersion
            <ns2:Fees>None</ns2:Fees>
            <ns2:AccessConstraints>none</ns2:AccessConstraints
         </ns2:ServiceIdentification>
         <ns2:ServiceProvider>
            <ns2:ProviderName>Meteorological Development Labora
            <ns2:ServiceContact>
               <ns2:IndividualName>Steve Olson</ns2:IndividualN
               <ns2:PositionName>MDL Nextgen POC</ns2:PositionN
               <ns2:ContactInfo>
                  <ns2:Phone>
                     <ns2:Voice>301-713-0056</ns2:Voice>
                  </ns2:Phone>
                  <ns2:Address>
                     <ns2:DeliveryPoint>W/OST24</ns2:DeliveryPo
                     <ns2:DeliveryPoint>1325 East-West Highway<
                     <ns2:City>Silver Spring</ns2:City>
                     <ns2:AdministrativeArea>Maryland</ns2:Admi
                     <ns2:PostalCode>20910</ns2:PostalCode>
                     <ns2:Country>U.S.</ns2:Country>
                     <ns2:ElectronicMailAddress>Steve[dot]R[do
                  </ns2:Address>
                  <ns2:OnlineResource ns4:href="http://www.wea
                  <ns2:HoursOfService>8x5x365</ns2:HoursOfServic
                  <ns2:ContactInstructions>email</ns2:ContactIn
               </ns2:ContactInfo>
               <ns2:Role>Lead Engineer</ns2:Role>
            </ns2:ServiceContact>
         </ns2:ServiceProvider>
         <ns2:OperationsMetadata>
            <ns2:Operation name="GetCapabilities">
               <ns2:DCP>
                  <ns2:HTTP>
                     <ns2:Post ns4:href="http://0.0.0.0:8887/w
                        <ns2:Constraint name="SOAP"/>
                     </ns2:Post>
                  </ns2:HTTP>
               </ns2:DCP>
               <ns2:Parameter name="service">
                  <ns2:AllowedValues>
                     <ns2:Value>WCS</ns2:Value>
                  </ns2:AllowedValues>
               </ns2:Parameter>
               <ns2:Parameter name="request">
                  <ns2:AllowedValues>
                     <ns2:Value>GetCapabilities</ns2:Value>
                  </ns2:AllowedValues>
               </ns2:Parameter>
```

Request Properties

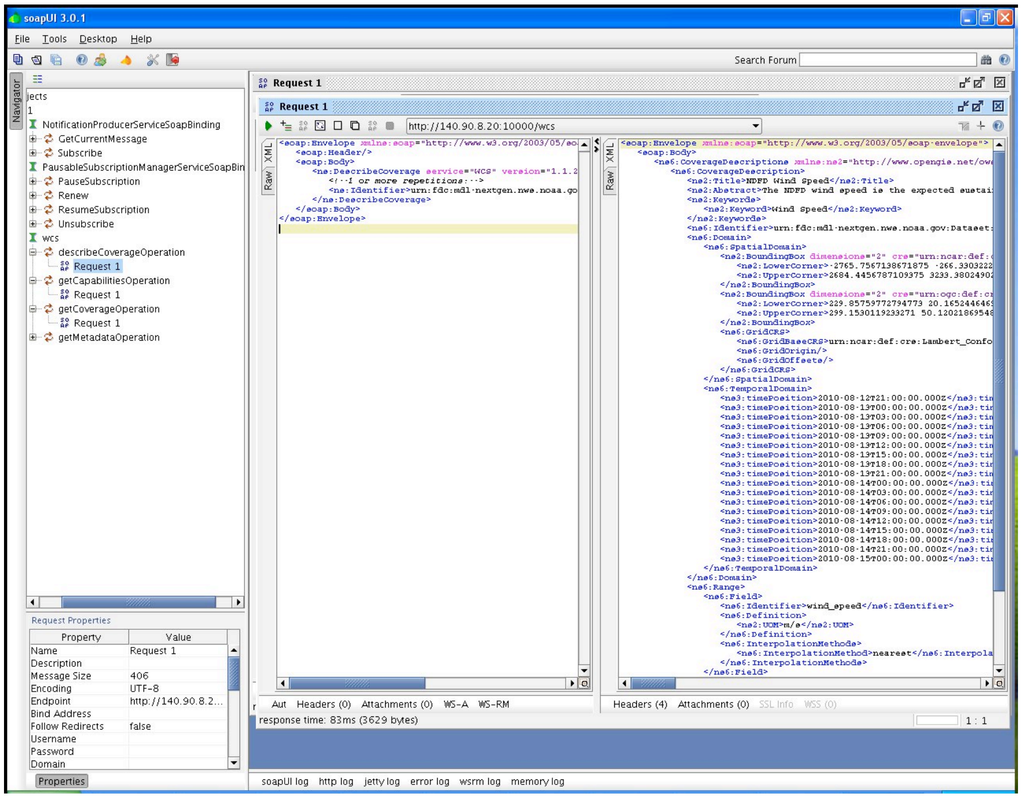
Property	Value
Name	Request 1
Description	
Message Size	333
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

response time: 2467ms (51179 bytes)

5. Step two: Perform a describeCoverage SOAP request using soapUI:
   - Double click on the “SOAP Request 1” associated with the describeCoverageOption. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server.
   - Cut and paste the following information in the left-hand pane, and click on green arrow, or replace the “**?**” with “**`urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDFD-CONUS-Wind-Speed`**”:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-
envelope" xmlns:ns="http://www.opengis.net/wcs/1.1">
   <soap:Header/>
   <soap:Body>
      <ns:DescribeCoverage service="WCS" version="1.1.2">
         <!--1 or more repetitions:-->
         <ns:Identifier>urn:fdc:mdl-
nextgen.nws.noaa.gov:Dataset:NDFD-CONUS-Wind-
Speed</ns:Identifier>
      </ns:DescribeCoverage>
   </soap:Body>
</soap:Envelope>
```

The screen should look something like this:



http://140.90.8.20:10000/wcs

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/so
   <soap:Header/>
   <soap:Body>
      <ns:DescribeCoverage service="WCS" version="1.1.2
         <!--1 or more repetitions:-->
         <ns:Identifier>urn:fdc:mdl-nextgen.nws.noaa.go
      </ns:DescribeCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Body>
      <ns6:CoverageDescriptions xmlns:ns2="http://www.opengis.net/ows
         <ns6:CoverageDescription>
            <ns2:Title>NDFD Wind Speed</ns2:Title>
            <ns2:Abstract>The NDFD wind speed is the expected sustai
            <ns2:Keywords>
               <ns2:Keyword>Wind Speed</ns2:Keyword>
            </ns2:Keywords>
            <ns6:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:
            <ns6:Domain>
               <ns6:SpatialDomain>
                  <ns2:BoundingBox dimensions="2" crs="urn:ncar:def:
                     <ns2:LowerCorner>-2765.7567138671875 -266.3303222
                     <ns2:UpperCorner>2684.4456787109375 3233.3802490
                  </ns2:BoundingBox>
                  <ns2:BoundingBox dimensions="2" crs="urn:ogc:def:cr
                     <ns2:LowerCorner>229.85759772794773 20.1652446469
                     <ns2:UpperCorner>299.1530119233271 50.12021869548
                  </ns2:BoundingBox>
                  <ns6:GridCRS>
                     <ns6:GridBaseCRS>urn:ncar:def:crs:Lambert_Confo
                     <ns6:GridOrigin/>
                     <ns6:GridOffsets/>
                  </ns6:GridCRS>
               </ns6:SpatialDomain>
               <ns6:TemporalDomain>
                  <ns3:timePosition>2010-08-12T21:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T00:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T03:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T06:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T09:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T12:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T15:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T18:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T21:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T00:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T03:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T06:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T09:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T12:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T15:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T18:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T21:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-15T00:00:00.000Z</ns3:tim
               </ns6:TemporalDomain>
            </ns6:Domain>
            <ns6:Range>
               <ns6:Field>
                  <ns6:Identifier>wind_speed</ns6:Identifier>
                  <ns6:Definition>
                     <ns2:UOM>m/s</ns2:UOM>
                  </ns6:Definition>
                  <ns6:InterpolationMethods>
                     <ns6:InterpolationMethod>nearest</ns6:Interpola
                  </ns6:InterpolationMethods>
               </ns6:Field>
```

response time: 83ms (3629 bytes)

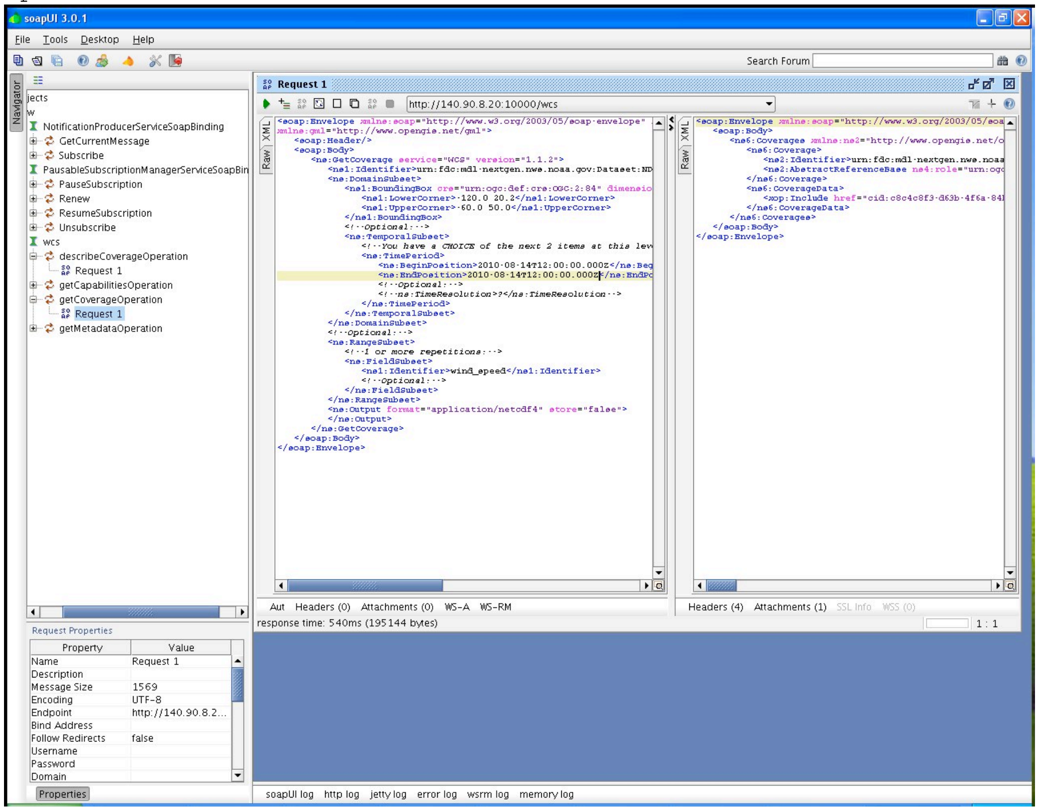
Request Properties

Property	Value
Name	Request 1
Description	
Message Size	406
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

6. Step three: Perform a getCoverage SOAP request using soapUI:
   - Double click on the “SOAP Request 1” associated with the getCoverageOption. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server.
   - Next, we will need to populate the correct getCoverage information in the left-side window pane. The easiest way to accomplish this is through a simple cut and paste operation. Cut and paste the following information in the left-hand pane. There is one field that will need to be adjusted, and that is the beginPosition and endPosition times. From the describeCoverage info, select the starting (beginPosition) and ending (endPosition) times you’d like to receive data for, the click on green arrow:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-
envelope" xmlns:ns="http://www.opengis.net/wcs/1.1"
xmlns:ns1="http://www.opengis.net/ows/1.1"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-
nextgen.nws.noaa.gov:Dataset:NDFD-CONUS-Wind-
Speed</ns1:Identifier>
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84"
dimensions="2">
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
```

```
                    <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
                 </ns1:BoundingBox>
                 <!--Optional:-->
                 <ns:TemporalSubset>
                    <!--You have a CHOICE of the next 2 items at this
level-->
                    <ns:TimePeriod>
                       <ns:BeginPosition>2010-08-
04T00:00:00.000Z</ns:BeginPosition>
                       <ns:EndPosition>2010-08-
04T00:00:00.000Z</ns:EndPosition>
                       <!--Optional:-->
                       <!--ns:TimeResolution>?</ns:TimeResolution-->
                    </ns:TimePeriod>
                 </ns:TemporalSubset>
              </ns:DomainSubset>
              <!--Optional:-->
              <ns:RangeSubset>
                 <!--1 or more repetitions:-->
                 <ns:FieldSubset>
                    <ns1:Identifier>wind_speed</ns1:Identifier>
                    <!--Optional:-->
                 </ns:FieldSubset>
              </ns:RangeSubset>
              <ns:Output format="application/netcdf4" store="false">
              </ns:Output>
           </ns:GetCoverage>
        </soap:Body>
     </soap:Envelope>
```



```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:ND
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84" dimensio
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this lev
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-14T12:00:00.000Z</ns:Beg
                  <ns:EndPosition>2010-08-14T12:00:00.000Z</ns:EndPo
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>
               <ns1:Identifier>wind_speed</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soa
   <soap:Body>
      <ns6:Coverages xmlns:ns2="http://www.opengis.net/o
         <ns6:Coverage>
            <ns2:Identifier>urn:fdc:mdl-nextgen.nws.noaa
            <ns2:AbstractReferenceBase ns4:role="urn:ogc
         </ns6:Coverage>
         <ns6:CoverageData>
            <xop:Include href="cid:c8c4c8f3-d63b-4f6a-84
         </ns6:CoverageData>
      </ns6:Coverages>
   </soap:Body>
</soap:Envelope>
```

Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (1) SSL Info WSS (0)

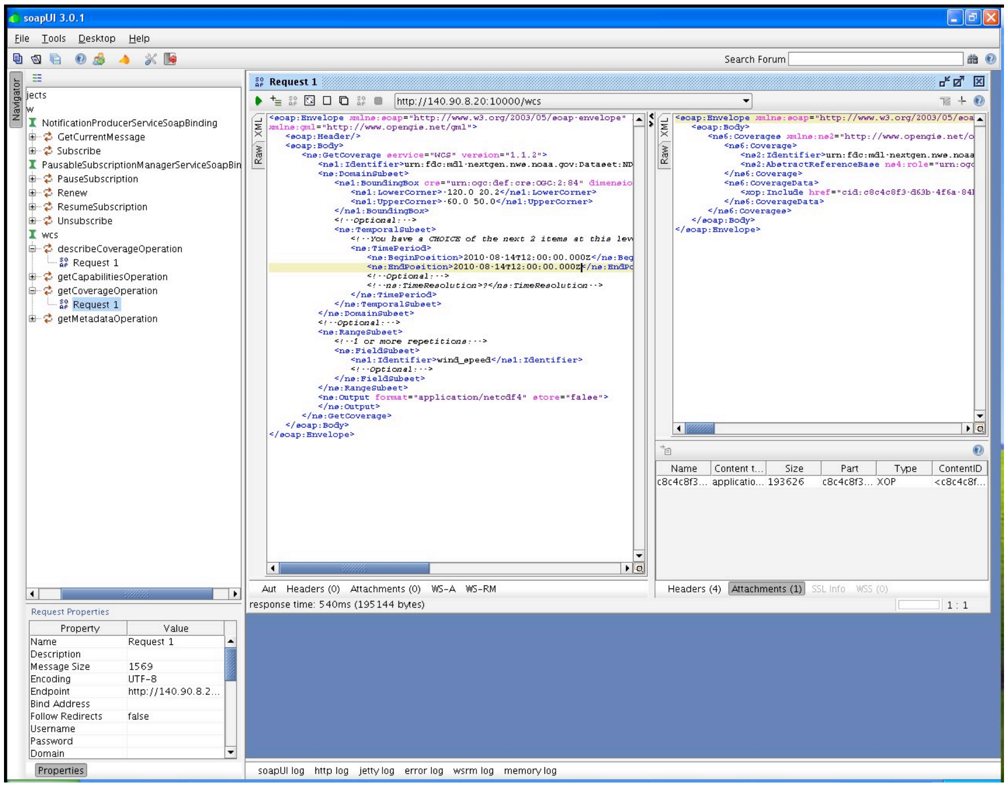
response time: 540ms (195144 bytes)

1 : 1

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

7. After execution, a getCoverage soap response will appear in the right-hand pane showing whether or not there was a successful request. If successful, there should be an attachment containing a netCDF4 file. Look for the word "Attachments" towards to lower left-hand corner of the getCoverage soap response (right-side pane).
8. Click once on the word "Attachments". A panel with "Name", "Content type", etc will appear. Below that line is the netCDF4 file.



http://140.90.8.20:10000/wcs

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:ND
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84" dimensio
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this lev
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-14T12:00:00.000Z</ns:Beg
                  <ns:EndPosition>2010-08-14T12:00:00.000Z</ns:EndPo
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>
               <ns1:Identifier>wind_speed</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soa
   <soap:Body>
      <ns6:Coverages xmlns:ns2="http://www.opengis.net/o
         <ns6:Coverage>
            <ns2:Identifier>urn:fdc:mdl-nextgen.nws.noaa
            <ns2:AbstractReferenceBase ns4:role="urn:ogc
         </ns6:Coverage>
         <ns6:CoverageData>
            <xop:Include href="cid:c8c4c8f3-d63b-4f6a-84
         </ns6:CoverageData>
      </ns6:Coverages>
   </soap:Body>
</soap:Envelope>
```

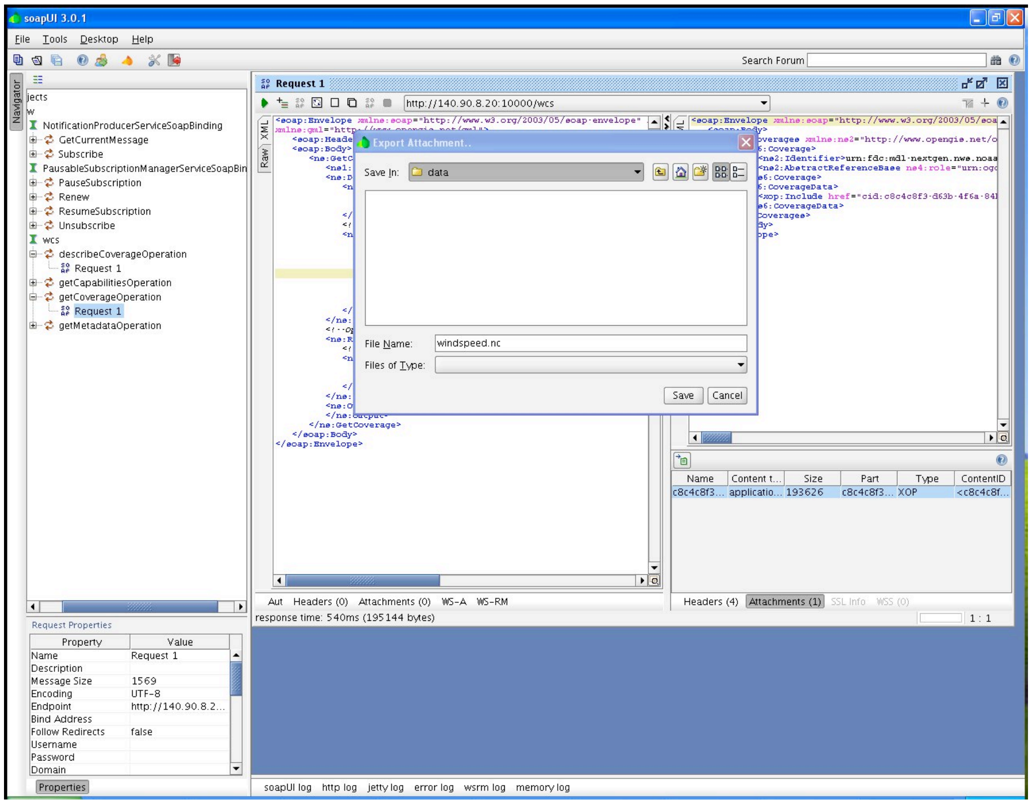
Name	Content t...	Size	Part	Type	ContentID
c8c4c8f3...	applicatio...	193626	c8c4c8f3...	XOP	<c8c4c8f...

response time: 540ms (195144 bytes)

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

9. Click once anywhere on this line. A green tab will appear that allows for the export of the selected attachment to a file. Click that tab once, and an "Export Attachment" GUI will appear. Save the file to a desired location.


soapUI 3.0.1
File Tools Desktop Help
Search Forum
Navigator
NotificationProducerServiceSoapBinding
GetCurrentMessage
Subscribe
PausableSubscriptionManagerServiceSoapBin
PauseSubscription
Renew
ResumeSubscription
Unsubscribe
wcs
describeCoverageOperation
Request 1
getCapabilitiesOperation
getCoverageOperation
Request 1
getMetadataOperation
Request 1
XML
Raw
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (1) SSL Info WSS (0)
soapUI log http log jetty log error log wsrm log memory log
Properties

http://140.90.8.20:10000/wcs

Export Attachment..

Save In: data

File Name: windspeed.nc

Files of Type:

Save Cancel

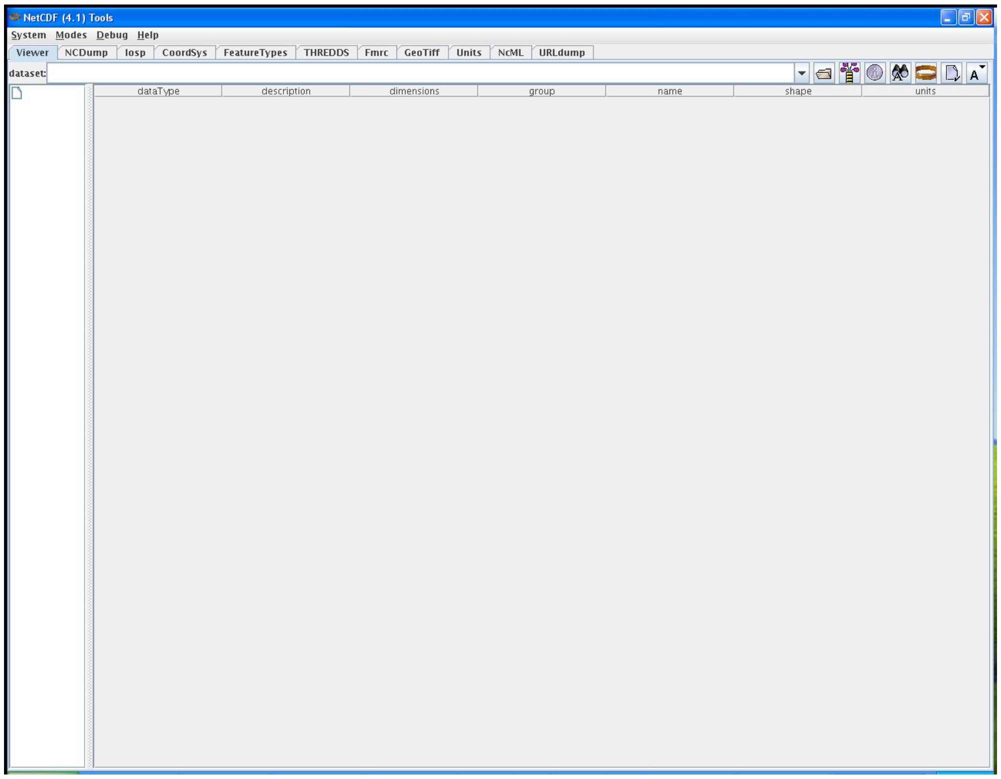
Request Properties

Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

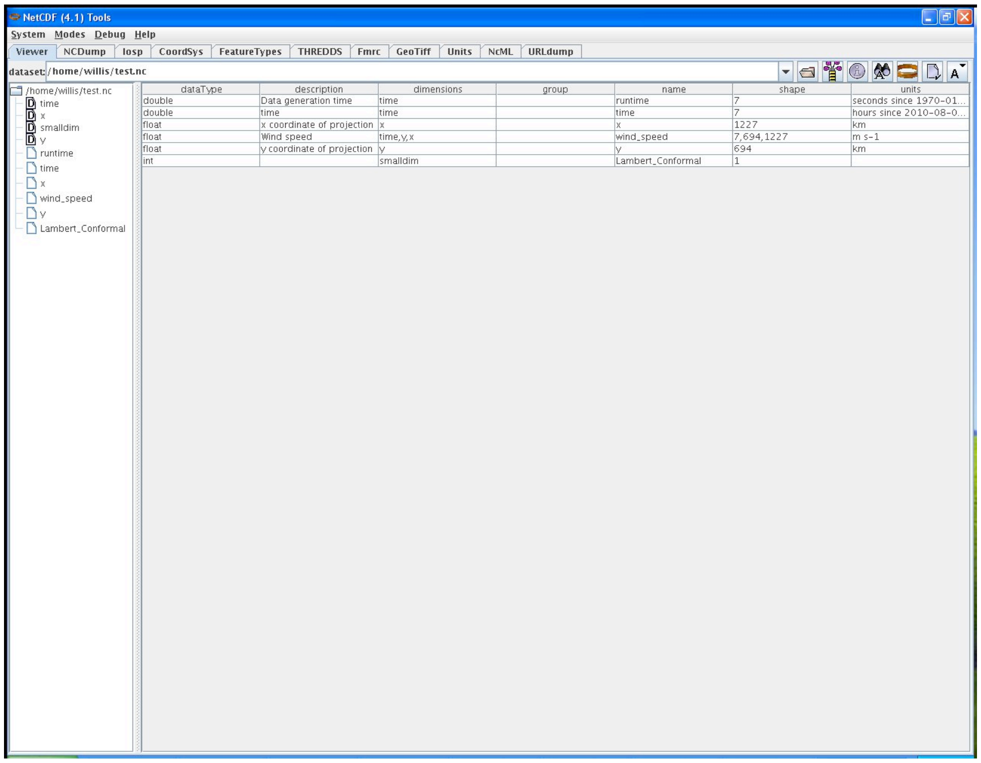
Name	Content t...	Size	Part	Type	ContentID
c8c4c8f3...	applicatio...	193626	c8c4c8f3...	XOP	<c8c4c8f...

response time: 540ms (195144 bytes)

10. A number of programs can be used to view the netCDF4 file, including IDV, toolsUI, ncview and ncdump. We have been using toolsUI version 4.1 (i.e. toolsUI-4.1.jar)
11. Start up toolsUI 4.1. The interface will look something like:


NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump
dataset:
dataType description dimensions group name shape units

12. Click on the open folder icon. Select the image you want to view. Click open, and the toolsUI GUI will look something like:

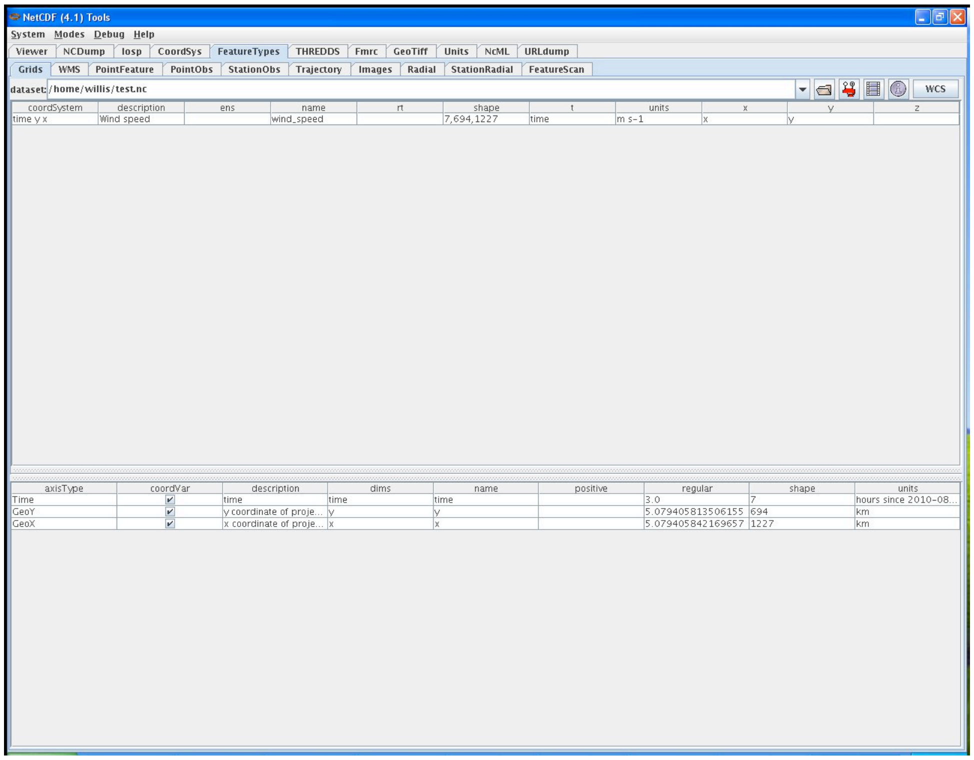

NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump

dataset: /home/willis/test.nc

```
/home/willis/test.nc
  time
  x
  smalldim
  y
  runtime
  time
  x
  wind_speed
  y
  Lambert_Conformal
```

dataType	description	dimensions	group	name	shape	units
double	Data generation time	time		runtime	7	seconds since 1970-01...
double	time	time		time	7	hours since 2010-08-0...
float	x coordinate of projection	x		x	1227	km
float	Wind speed	time,y,x		wind_speed	7,694,1227	m s-1
float	y coordinate of projection	y		y	694	km
int		smalldim		Lambert_Conformal	1	

13. The viewer tab will describe information about the file. In this case, its NDFD wind speed.
14. To view the image, click on the FeatureTypes tab, and click on the open folder icon again. Select the image you want to view (test.nc in our example) and click open. The toolsUI GUI will look something like:

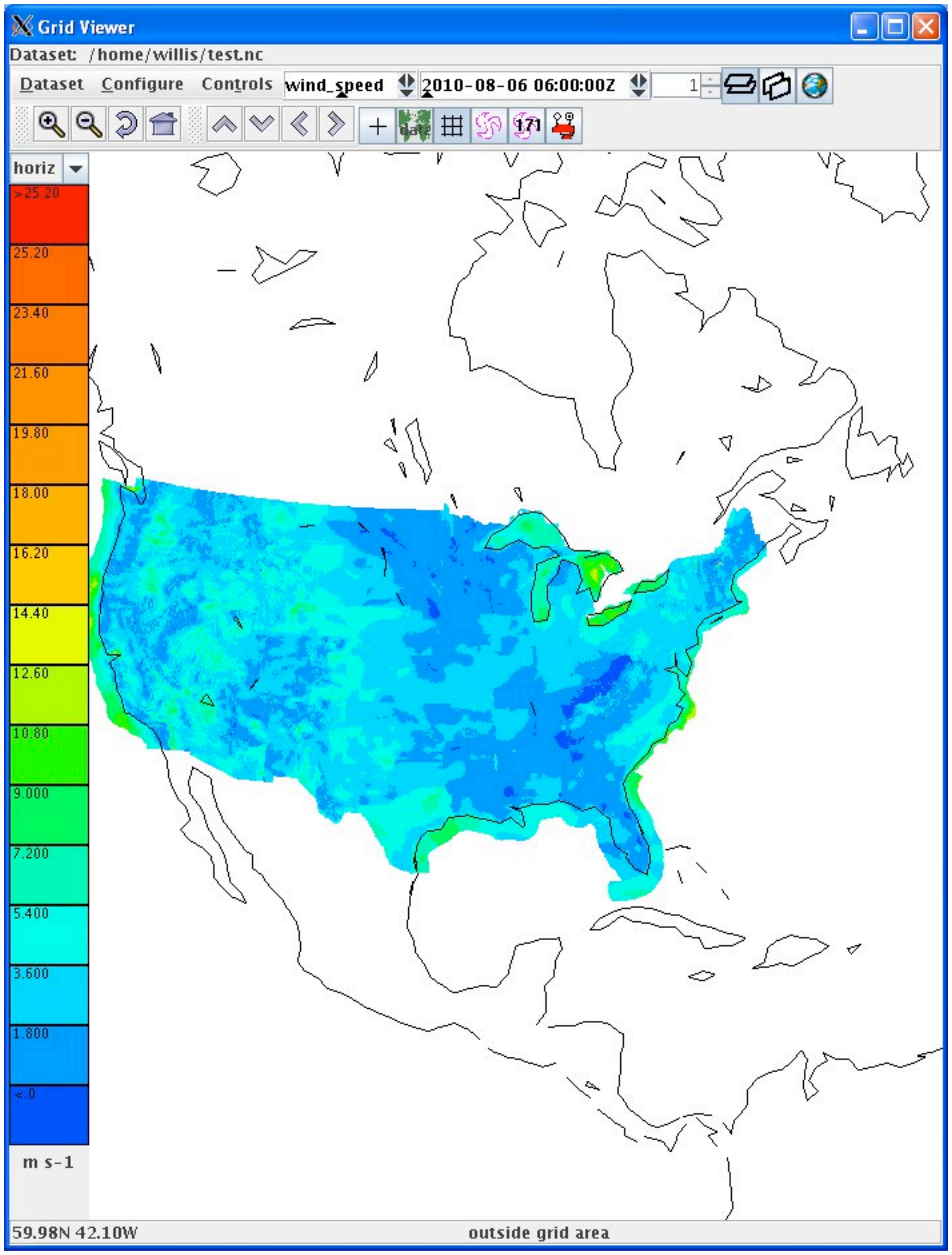

NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump
Grids WMS PointFeature PointObs StationObs Trajectory Images Radial StationRadial FeatureScan
WCS

dataset: /home/willis/test.nc

coordSystem	description	ens	name	rt	shape	t	units	x	y	z
time y x	Wind speed		wind_speed		7,694,1227	time	m s-1	x	y	

axisType	coordVar	description	dims	name	positive	regular	shape	units
Time	✓	time	time	time		3.0	7	hours since 2010-08...
GeoY	✓	y coordinate of proje...	y	y		5.079405813506155	694	km
GeoX	✓	x coordinate of proje...	x	x		5.079405842169657	1227	km

15. Click on the wind speed tab and then click the red viewer icon. A new GUI window will appear. Initially, the viewer window should be completely black. Clicking on the red icon one more time will display the image. The viewer GUI should look something like:



8.1.4.4 *Test Data Reduction and Analysis*

All data collected automatically from the PC during testing will be archived and stored for future reference. All issues that arise during the test will be documented in a Problem Tracking Report (PTR) summarizing the description, criticality and proposed solution of the issue.

8.2 NDFD Convective Hazard Outlook dataset

8.2.1 Introduction

8.2.1.1 Purpose and Scope

The purpose of this test is to demonstrate the way the 4-D Wx Data Cube may operate with National Digital Forecast Database (NDFD) Wind Speed data at initial operating capacity. The publishing and discovery of metadata about the WCS service and the NDFD Wind Speed dataset itself from federated reg/rep agency repositories will be tested. The requirements of the WCS RI are located in Appendix C of the Capability Evaluation Plan. System level requirements in Appendix D. Latencies, although not required, can be measured.

8.2.2 Reference Documents

1. A NWS NDFD program description can be found at http://www.nws.noaa.gov/ndfd/.
2. NDFD definitions can be found at http://www.nws.noaa.gov/ndfd/definitions.htm
3. The SPC Convective Hazard Outlook page can be found at: http://www.spc.ncep.noaa.gov/products/outlook/
4. A description of how to interpret the day1, day2 and day3 outlooks can be found at: http://www.spc.noaa.gov/misc/SPC_probotlk_info.html

8.2.3 Test Description

8.2.3.1 System Under Test

MDL will be the source system which will be providing NDFD/NDGD data via NOAAnet

8.2.3.2 Test Setup

Prior to the test, the evaluation PC must be loaded with RedHat 5 O/S and configured with OGC compliant software to request and display weather data from the Cube. Unidata's IDV must also be installed to verify that the dataset is available.

a. Test Equipment at the Tech Center

NAS Simulator, evaluation PC running the IDV, GSD's Testing Portal

b. Test Equipment at remote site

MDL's WCSRI endpoint with live data

8.2.3.3 Personnel

FAA WJHTC, MDL, GSD

8.2.3.4 Client Location

NWEC lab at WJHTC

8.2.3.5 Server Location

NOAA/NWS/MDL

8.2.4 Test Conduct

8.2.4.1 Security Concerns

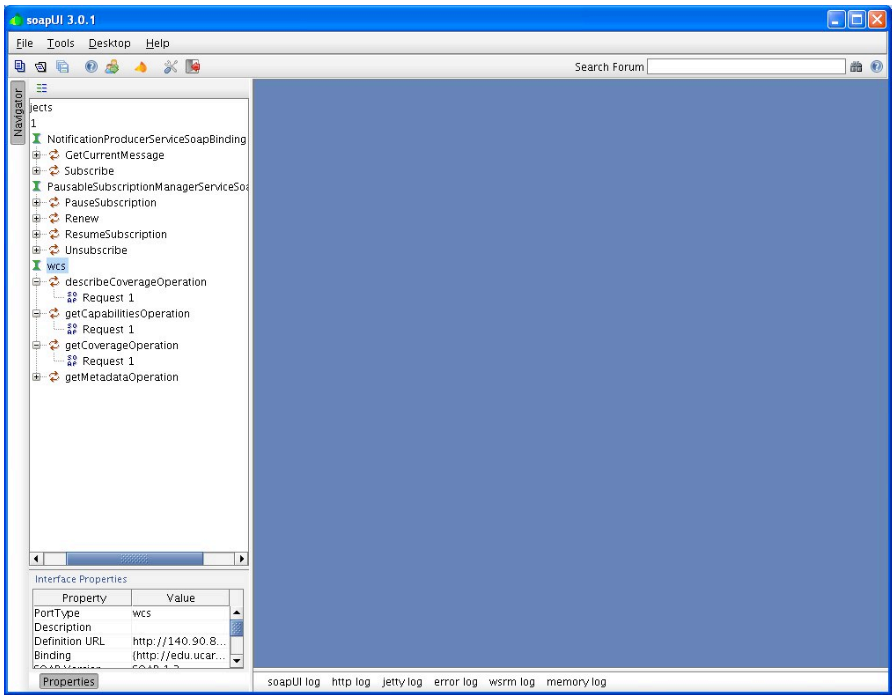
Performance on operational system – to get around this do it at 19Z. Least impact to FTI. Compliant with NIST guided, NOAA/FAA policies on security architecture and relevant information security technologies through a XML gateway.

8.2.4.2 Requirements Under Test

- RI shall support all WCSRI 1.1.0/2.0 spec requirements in Appendix C of the current FY10 Capability Evaluation Plan
- Though not required, the system performance (e.g. latency) shall be measured/baselined.

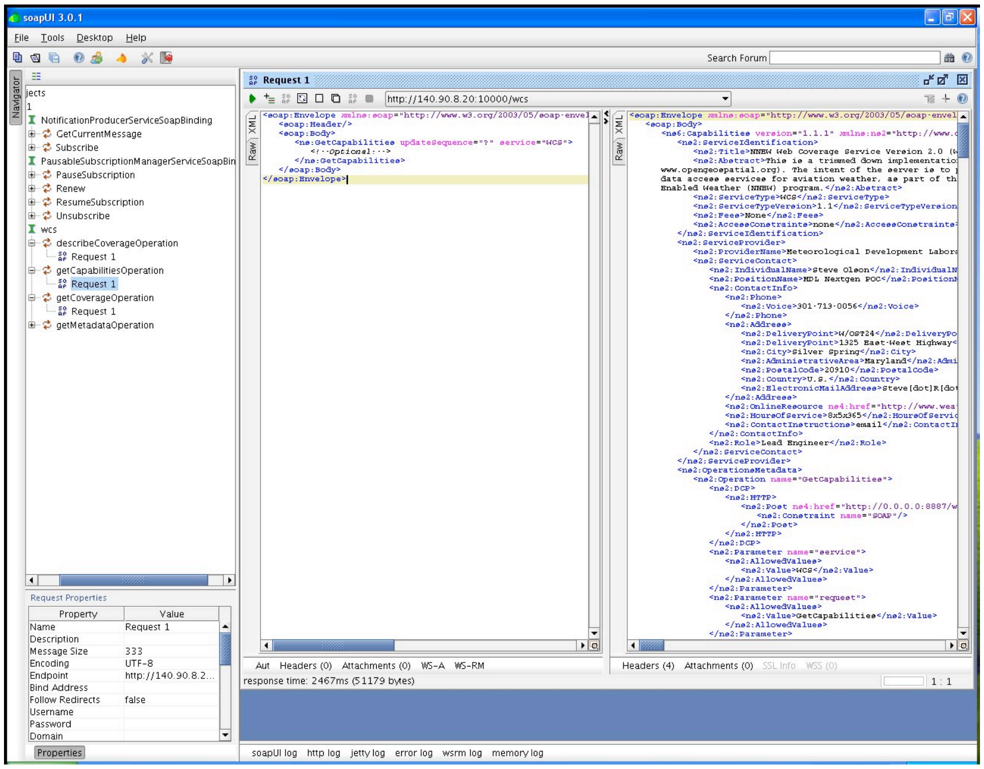
8.2.4.3 Procedures

1. Start up soapUI version 3.0+.
   - Import mdl-wcs-soapui-project.xml file if needed. Click on **File**, then **Import Project,** then select file **mdl-wcs-soapui-project.xml , and click on Open**.
2. Click on the “plus” sign and this will expand the **wcs** to include the following SOAP functions:
   - **describeCoverageOperation**
   - **getCapabilitiesOperation**
   - **getCoverageOperation**
   - **getMetadataOperation**



Property	Value
PortType	wcs
Description	
Definition URL	http://140.90.8...
Binding	{http://edu.ucar...

3. Click the "+" symbol for each of these features, and you will reveal a "SOAP Request 1". These soap requests serve as the interface between the user, and our WCS (where the data resides).
4. Step one: Perform a getCapabilities SOAP request using soapUI:
   - Double click on the "SOAP Request 1" associated with getCapabilitiesOperation. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server. There are <u>no fields</u> that need to be populated with this request. This is a simple request to find out what data is available and more background information about our WCS.
   - Click on the green arrow (Submit request to a specified endpoint URL). After execution, a getCapabilities soap response will appear in the right-hand pane indicating some background information about who's offering the service, and what data is available via our WCS.
   - The most important information that is contained in this soap response is the "ns2:**parameter name="Identifier**". This will provide "Values" indicating what data is available. The "urn: value form this soap response is an important input to the second request.
   - Copy the urn information (**urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDFD-CONUS-Convective-Hazard-Outlook**) into the copy and paste buffer.



soapUI 3.0.1

Request 1

http://140.90.8.20:10000/wcs

```
jects
1
NotificationProducerServiceSoapBinding
    GetCurrentMessage
    Subscribe
PausableSubscriptionManagerServiceSoapBin
    PauseSubscription
    Renew
    ResumeSubscription
    Unsubscribe
wcs
    describeCoverageOperation
        Request 1
    getCapabilitiesOperation
        Request 1
    getCoverageOperation
        Request 1
    getMetadataOperation
```

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	333
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envel
   <soap:Header/>
   <soap:Body>
      <ns:GetCapabilities updateSequence="?" service="WCS">
         <!--Optional:-->
      </ns:GetCapabilities>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envel
   <soap:Body>
      <ns6:Capabilities version="1.1.1" xmlns:ns2="http://www.
         <ns2:ServiceIdentification>
            <ns2:Title>NNEW Web Coverage Service Version 2.0 (W
            <ns2:Abstract>This is a trimmed down implementatio
      www.opengeospatial.org). The intent of the server is to p
      data access services for aviation weather, as part of th
      Enabled Weather (NNEW) program.</ns2:Abstract>
            <ns2:ServiceType>WCS</ns2:ServiceType>
            <ns2:ServiceTypeVersion>1.1</ns2:ServiceTypeVersion
            <ns2:Fees>None</ns2:Fees>
            <ns2:AccessConstraints>none</ns2:AccessConstraints
         </ns2:ServiceIdentification>
         <ns2:ServiceProvider>
            <ns2:ProviderName>Meteorological Development Labora
            <ns2:ServiceContact>
               <ns2:IndividualName>Steve Olson</ns2:IndividualN
               <ns2:PositionName>MDL Nextgen POC</ns2:PositionN
               <ns2:ContactInfo>
                  <ns2:Phone>
                     <ns2:Voice>301-713-0056</ns2:Voice>
                  </ns2:Phone>
                  <ns2:Address>
                     <ns2:DeliveryPoint>W/OST24</ns2:DeliveryPo
                     <ns2:DeliveryPoint>1325 East-West Highway<
                     <ns2:City>Silver Spring</ns2:City>
                     <ns2:AdministrativeArea>Maryland</ns2:Admi
                     <ns2:PostalCode>20910</ns2:PostalCode>
                     <ns2:Country>U.S.</ns2:Country>
                     <ns2:ElectronicMailAddress>Steve[dot]R[dot
                  </ns2:Address>
                  <ns2:OnlineResource ns4:href="http://www.wea
                  <ns2:HoursOfService>8x5x365</ns2:HoursOfServic
                  <ns2:ContactInstructions>email</ns2:ContactIn
               </ns2:ContactInfo>
               <ns2:Role>Lead Engineer</ns2:Role>
            </ns2:ServiceContact>
         </ns2:ServiceProvider>
         <ns2:OperationsMetadata>
            <ns2:Operation name="GetCapabilities">
               <ns2:DCP>
                  <ns2:HTTP>
                     <ns2:Post ns4:href="http://0.0.0.0:8887/w
                        <ns2:Constraint name="SOAP"/>
                     </ns2:Post>
                  </ns2:HTTP>
               </ns2:DCP>
               <ns2:Parameter name="service">
                  <ns2:AllowedValues>
                     <ns2:Value>WCS</ns2:Value>
                  </ns2:AllowedValues>
               </ns2:Parameter>
               <ns2:Parameter name="request">
                  <ns2:AllowedValues>
                     <ns2:Value>GetCapabilities</ns2:Value>
                  </ns2:AllowedValues>
               </ns2:Parameter>
```

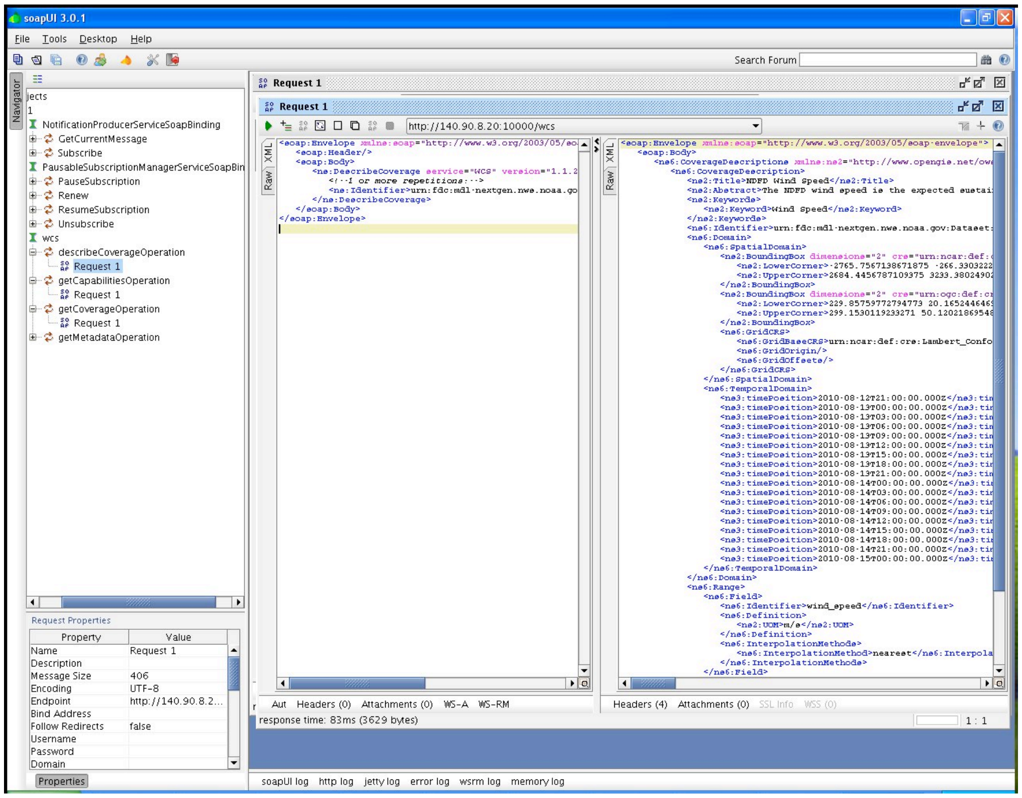
response time: 2467ms (51179 bytes)

File Tools Desktop Help
Search Forum
Navigator
Raw XML
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (0) SSL Info WSS (0)
1 : 1
Properties
soapUI log http log jetty log error log wsrm log memory log

5. Step two: Perform a describeCoverage SOAP request using soapUI:
   - Double click on the "SOAP Request 1" associated with the describeCoverageOption. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server.
   - Cut and paste the following information in the left-hand pane, and click on green arrow, or replace the "**?**" with "**urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDFD-CONUS-Convective-Hazard-Outlook**":

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-
envelope" xmlns:ns="http://www.opengis.net/wcs/1.1">
   <soap:Header/>
   <soap:Body>
      <ns:DescribeCoverage service="WCS" version="1.1.2">
         <!--1 or more repetitions:-->
  <ns:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDFD-
CONUS-Convective-Hazard-Outlook</ns:Identifier>
      </ns:DescribeCoverage>
   </soap:Body>
</soap:Envelope>
```

1. The screen should look something like this:



Property	Value
Name	Request 1
Description	
Message Size	406
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

```
http://140.90.8.20:10000/wcs
```

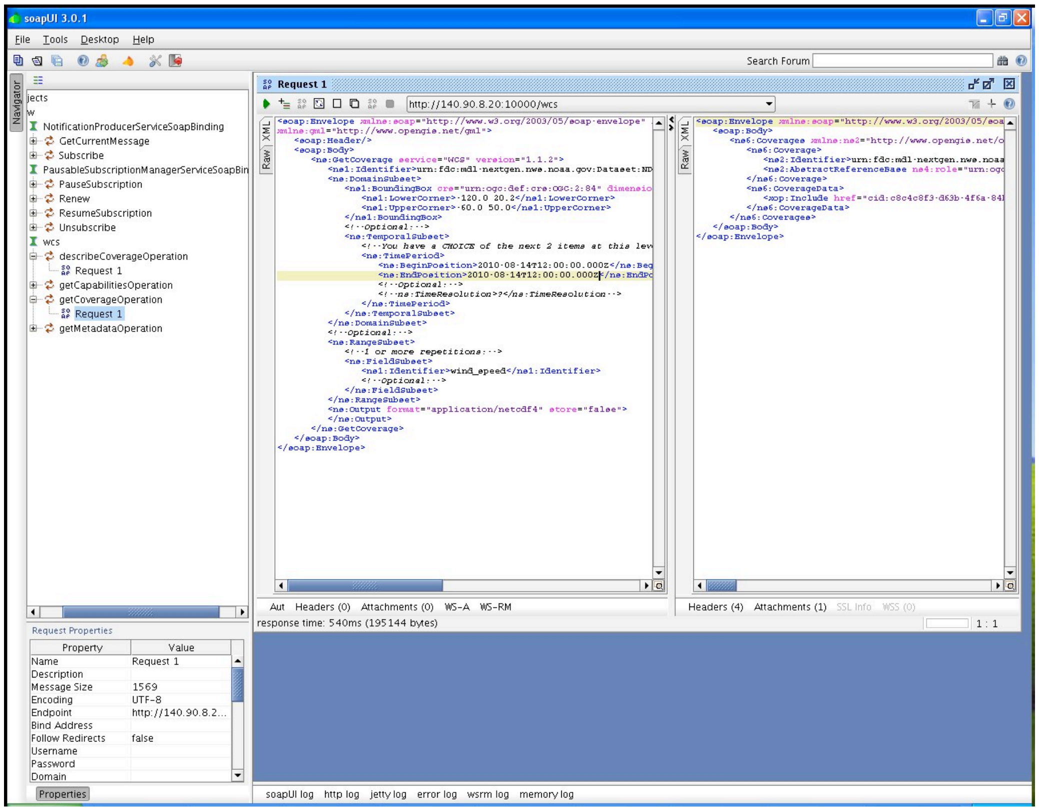
```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/so
   <soap:Header/>
   <soap:Body>
      <ns:DescribeCoverage service="WCS" version="1.1.2
         <!--1 or more repetitions:-->
         <ns:Identifier>urn:fdc:mdl-nextgen.nws.noaa.go
      </ns:DescribeCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Body>
      <ns6:CoverageDescriptions xmlns:ns2="http://www.opengis.net/ows
         <ns6:CoverageDescription>
            <ns2:Title>NDFD Wind Speed</ns2:Title>
            <ns2:Abstract>The NDFD wind speed is the expected sustai
            <ns2:Keywords>
               <ns2:Keyword>Wind Speed</ns2:Keyword>
            </ns2:Keywords>
            <ns6:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:
            <ns6:Domain>
               <ns6:SpatialDomain>
                  <ns2:BoundingBox dimensions="2" crs="urn:ncar:def:
                     <ns2:LowerCorner>-2765.7567138671875 -266.3303222
                     <ns2:UpperCorner>2684.4456787109375 3233.3802490
                  </ns2:BoundingBox>
                  <ns2:BoundingBox dimensions="2" crs="urn:ogc:def:cr
                     <ns2:LowerCorner>229.85759772794773 20.165244646
                     <ns2:UpperCorner>299.1530119233271 50.12021869548
                  </ns2:BoundingBox>
                  <ns6:GridCRS>
                     <ns6:GridBaseCRS>urn:ncar:def:crs:Lambert_Confo
                     <ns6:GridOrigin/>
                     <ns6:GridOffsets/>
                  </ns6:GridCRS>
               </ns6:SpatialDomain>
               <ns6:TemporalDomain>
                  <ns3:timePosition>2010-08-12T21:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T00:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T03:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T06:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T09:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T12:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T15:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T18:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T21:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T00:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T03:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T06:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T09:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T12:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T15:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T18:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T21:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-15T00:00:00.000Z</ns3:tin
               </ns6:TemporalDomain>
            </ns6:Domain>
            <ns6:Range>
               <ns6:Field>
                  <ns6:Identifier>wind_speed</ns6:Identifier>
                  <ns6:Definition>
                     <ns2:UOM>m/s</ns2:UOM>
                  </ns6:Definition>
                  <ns6:InterpolationMethods>
                     <ns6:InterpolationMethod>nearest</ns6:Interpola
                  </ns6:InterpolationMethods>
               </ns6:Field>
```

response time: 83ms (3629 bytes)

6. Step three: Perform a getCoverage SOAP request using soapUI:
   - Double click on the “SOAP Request 1” associated with the getCoverageOption. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server.
   - Next, we will need to populate the correct getCoverage information in the left-side window pane. The easiest way to accomplish this is through a simple cut and paste operation. Cut and paste the following information in the left-hand pane. There is one field that will need to be adjusted, and that is the beginPosition and endPosition times. From the describeCoverage info, select the starting (beginPosition) and ending (endPosition) times you’d like to receive data for, the click on green arrow:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns="http://www.opengis.net/wcs/1.1"
xmlns:ns1="http://www.opengis.net/ows/1.1"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDFD-
CONUS-Convective-Hazard-Outlook</ns1:Identifier>
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84"
dimensions="2">
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this level-
->
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-
04T00:00:00.000Z</ns:BeginPosition>
                  <ns:EndPosition>2010-08-
04T00:00:00.000Z</ns:EndPosition>
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>

<ns1:Identifier>convective_hazard_outlook</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```


soapUI 3.0.1
File Tools Desktop Help
Search Forum
Navigator
jects
w
NotificationProducerServiceSoapBinding
GetCurrentMessage
Subscribe
PausableSubscriptionManagerServiceSoapBin
PauseSubscription
Renew
ResumeSubscription
Unsubscribe
wcs
describeCoverageOperation
Request 1
getCapabilitiesOperation
getCoverageOperation
Request 1
getMetadataOperation
Request 1
http://140.90.8.20:10000/wcs
XML
Raw
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (1) SSL Info WSS (0)
1 : 1
Properties
soapUI log http log jetty log error log wsrm log memory log

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:ND
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84" dimensio
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this lev
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-14T12:00:00.000Z</ns:Beg
                  <ns:EndPosition>2010-08-14T12:00:00.000Z</ns:EndPo
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>
               <ns1:Identifier>wind_speed</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```

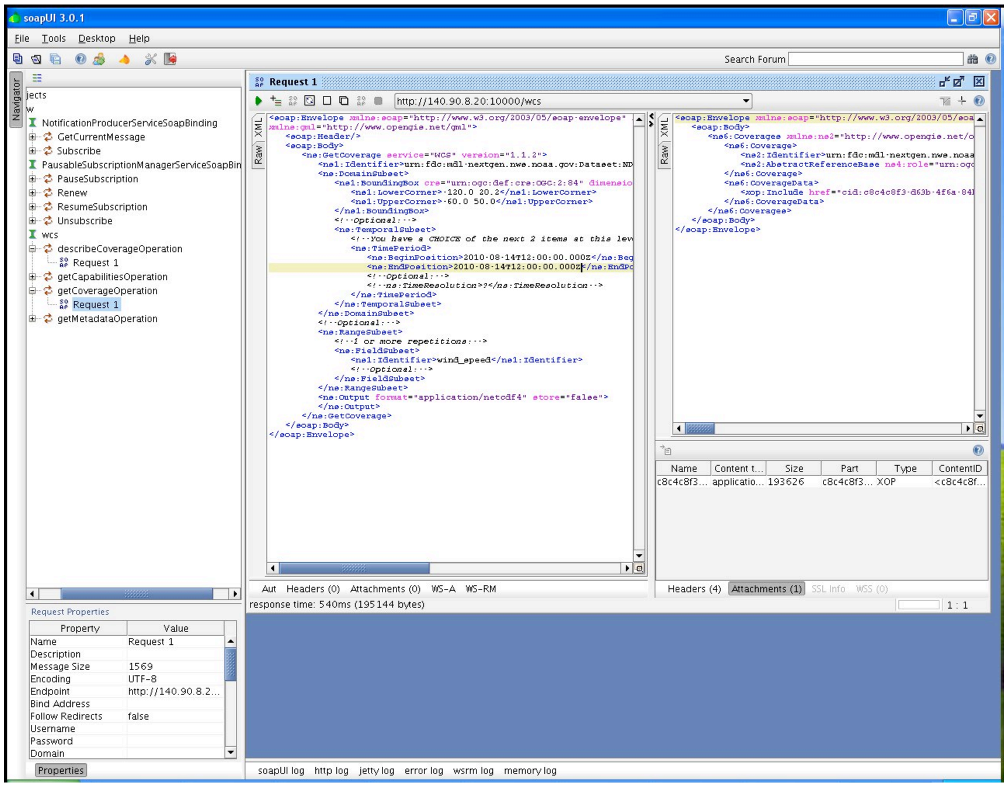
```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soa
   <soap:Body>
      <ns6:Coverages xmlns:ns2="http://www.opengis.net/o
         <ns6:Coverage>
            <ns2:Identifier>urn:fdc:mdl-nextgen.nws.noaa
            <ns2:AbstractReferenceBase ns4:role="urn:ogc
         </ns6:Coverage>
         <ns6:CoverageData>
            <xop:Include href="cid:c8c4c8f3-d63b-4f6a-84
         </ns6:CoverageData>
      </ns6:Coverages>
   </soap:Body>
</soap:Envelope>
```

response time: 540ms (195144 bytes)

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

7. After execution, a getCoverage soap response will appear in the right-hand pane showing whether or not there was a successful request. If successful, there should be an attachment containing a netCDF4 file. Look for the word “Attachments” towards to lower left-hand corner of the getCoverage soap response (right-side pane).
8. Click once on the word “Attachments”. A panel with “Name”, “Content type”, etc will appear. Below that line is the netCDF4 file.



```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:ND
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84" dimensio
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this lev
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-14T12:00:00.000Z</ns:Beg
                  <ns:EndPosition>2010-08-14T12:00:00.000Z</ns:EndPo
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>
               <ns1:Identifier>wind_speed</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soa
   <soap:Body>
      <ns6:Coverages xmlns:ns2="http://www.opengis.net/o
         <ns6:Coverage>
            <ns2:Identifier>urn:fdc:mdl-nextgen.nws.noaa
            <ns2:AbstractReferenceBase ns4:role="urn:ogc
         </ns6:Coverage>
         <ns6:CoverageData>
            <xop:Include href="cid:c8c4c8f3-d63b-4f6a-84
         </ns6:CoverageData>
      </ns6:Coverages>
   </soap:Body>
</soap:Envelope>
```

Name	Content t...	Size	Part	Type	ContentID
c8c4c8f3...	applicatio...	193626	c8c4c8f3...	XOP	<c8c4c8f...

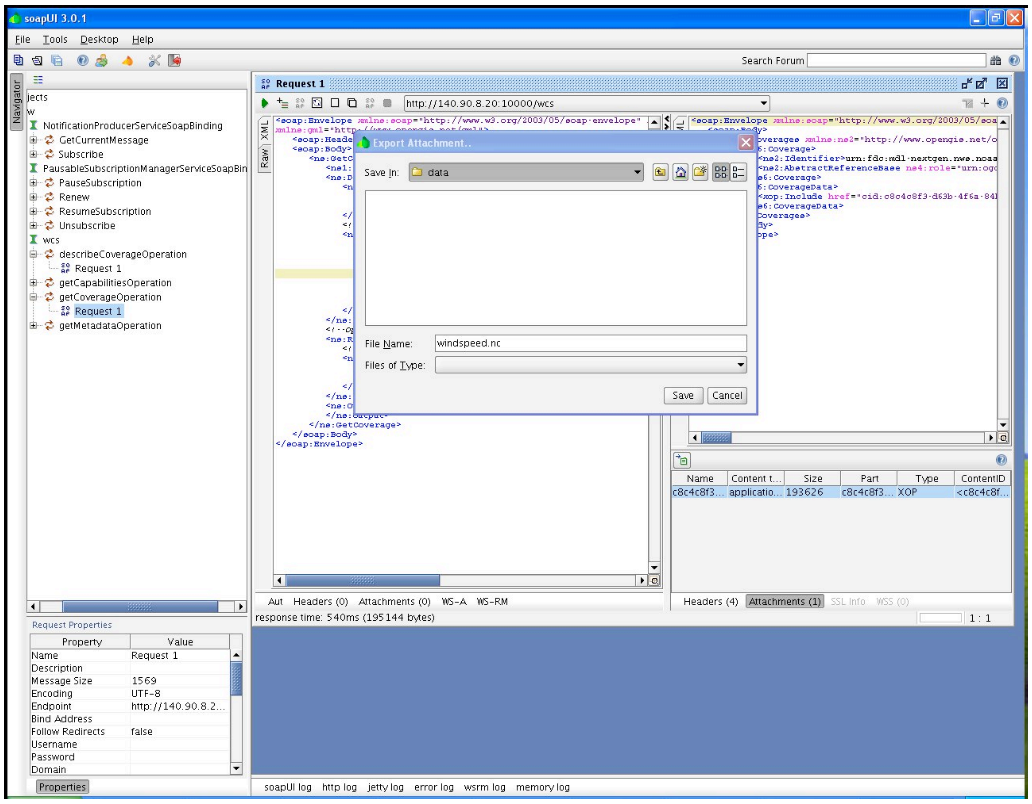
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (1) SSL Info WSS (0)

response time: 540ms (195144 bytes)

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

9. Click once anywhere on this line. A green tab will appear that allows for the export of the selected attachment to a file. Click that tab once, and an “Export Attachment” GUI will appear. Save the file to a desired location. A number of programs can be used to view the netCDF4 file, including ncview and ncdump.



soapUI 3.0.1

Request 1 — http://140.90.8.20:10000/wcs

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
        </ns:GetCoverage>
    </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soa
  overages xmlns:ns2="http://www.opengis.net/o
  6:Coverage>
  <ns2:Identifier>urn:fdc:mdl-nextgen.nws.noaa
  <ns2:AbstractReferenceBase ns4:role="urn:ogc
  s6:Coverage>
  6:CoverageData>
  <xop:Include href="cid:c8c4c8f3-d63b-4f6a-84
  s6:CoverageData>
  Coverages>
  dy>
  ope>
```

Export Attachment..

Save In: data

File Name: windspeed.nc

Files of Type:

Name	Content t...	Size	Part	Type	ContentID
c8c4c8f3...	applicatio...	193626	c8c4c8f3...	XOP	<c8c4c8f...

response time: 540ms (195144 bytes)

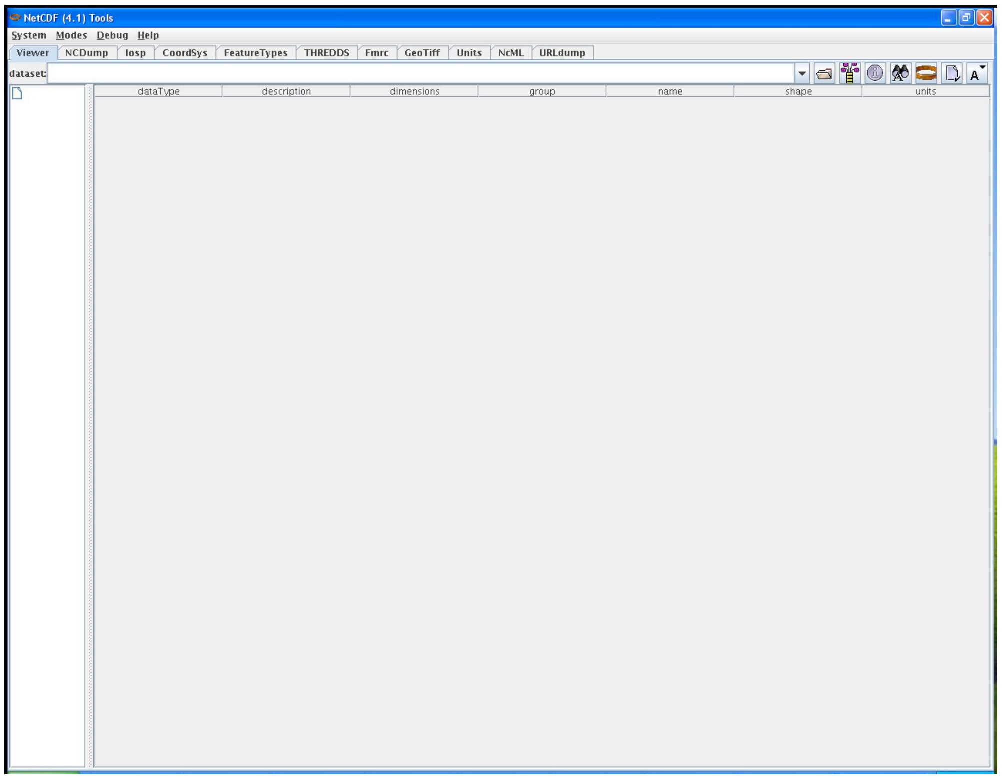
Request Properties

Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

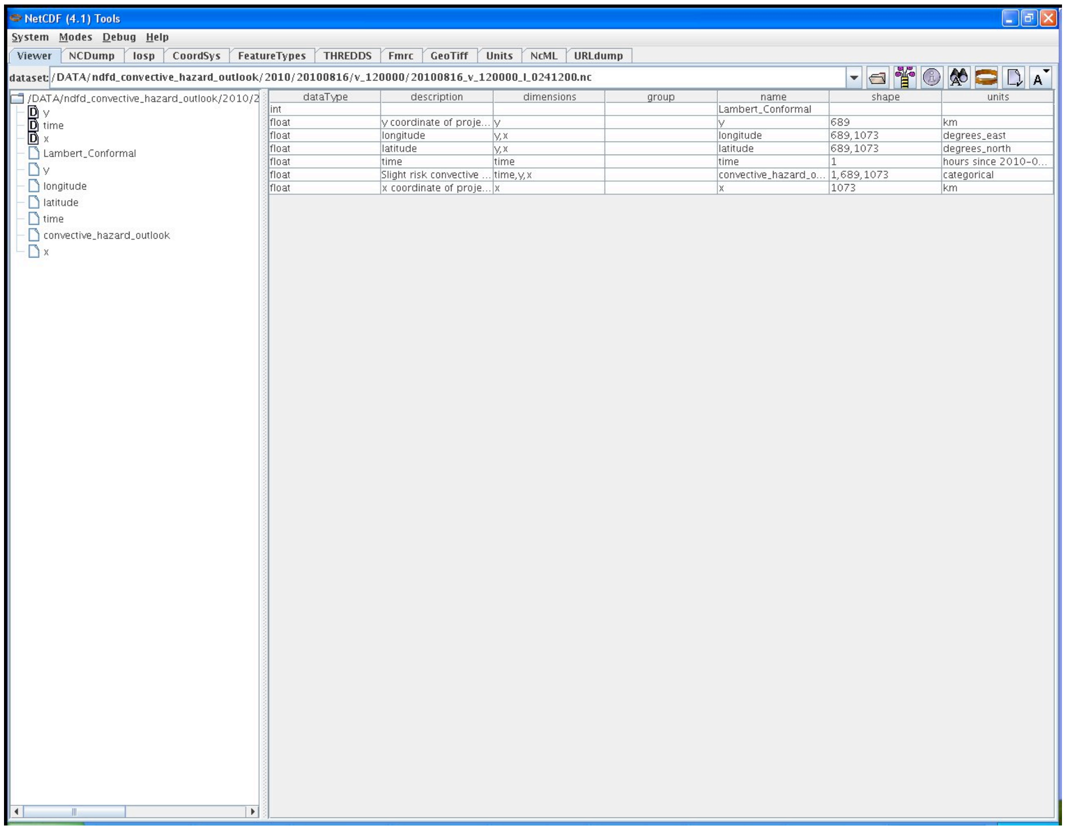
File Tools Desktop Help
Search Forum
Navigator
NotificationProducerServiceSoapBinding
GetCurrentMessage
Subscribe
PausableSubscriptionManagerServiceSoapBin
PauseSubscription
Renew
ResumeSubscription
Unsubscribe
wcs
describeCoverageOperation
Request 1
getCapabilitiesOperation
getCoverageOperation
Request 1
getMetadataOperation
XML
Raw
Save
Cancel
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (1) SSL Info WSS (0)
1 : 1
Properties
soapUI log http log jetty log error log wsrm log memory log

(**NOTE: Wind Speed is shown here**)

10. A number of programs can be used to view the netCDF4 file, including IDV, toolsUI, ncview and ncdump. We have been using toolsUI version 4.1 (i.e. toolsUI-4.1.jar)
11. Start up toolsUI 4.1. The interface will look something like:


NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump
dataset:
dataType description dimensions group name shape units

12. Click on the open folder icon. Select the image you want to view. Click open, and the toolsUI GUI will look something like:

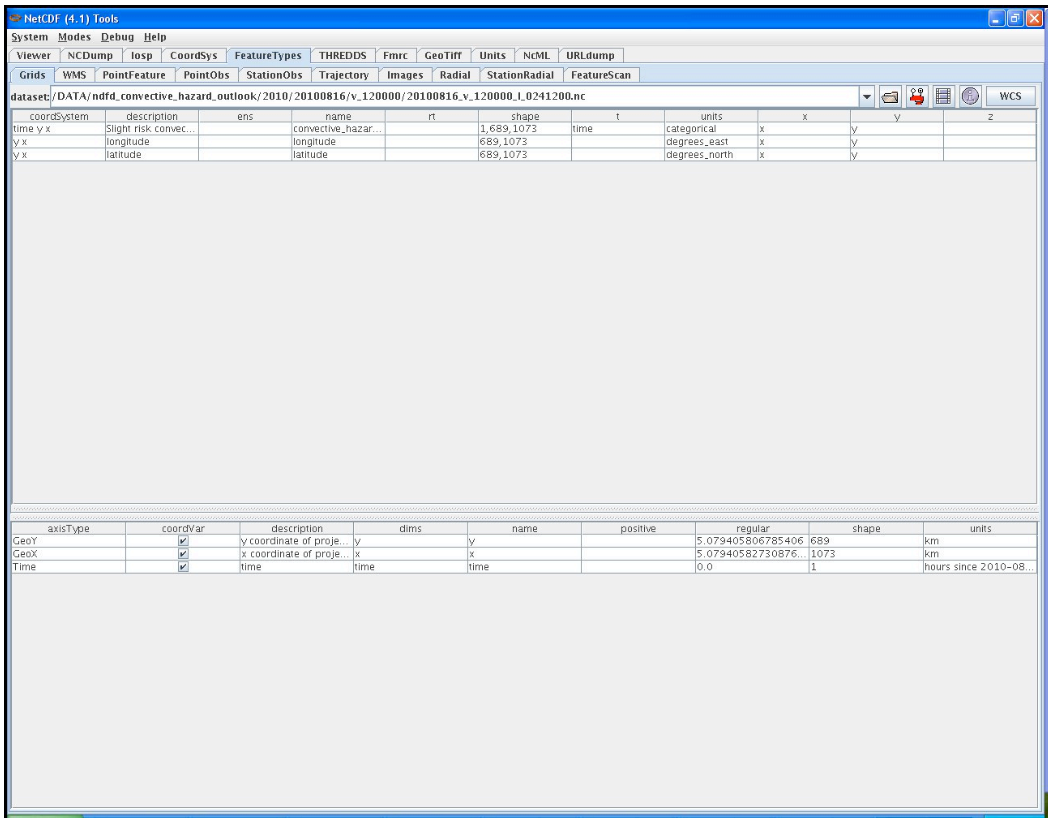

NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump

dataset: /DATA/ndfd_convective_hazard_outlook/2010/20100816/v_120000/20100816_v_120000_I_0241200.nc

```
/DATA/ndfd_convective_hazard_outlook/2010/2
  D y
  D time
  D x
    Lambert_Conformal
    y
    longitude
    latitude
    time
    convective_hazard_outlook
    x
```

dataType	description	dimensions	group	name	shape	units
int				Lambert_Conformal		
float	y coordinate of proje...	y		y	689	km
float	longitude	y,x		longitude	689,1073	degrees_east
float	latitude	y,x		latitude	689,1073	degrees_north
float	time	time		time	1	hours since 2010-0...
float	Slight risk convective ...	time,y,x		convective_hazard_o...	1,689,1073	categorical
float	x coordinate of proje...	x		x	1073	km

13. The viewer tab will describe information about the file. In this case, its NDFD convective hazard outlook.
14. To view the image, click on the FeatureTypes tab, and click on the open folder icon again. Select the image you want to view (20100816_v_120000_0241200.nc in our example) and click open. The toolsUI GUI will look something like:

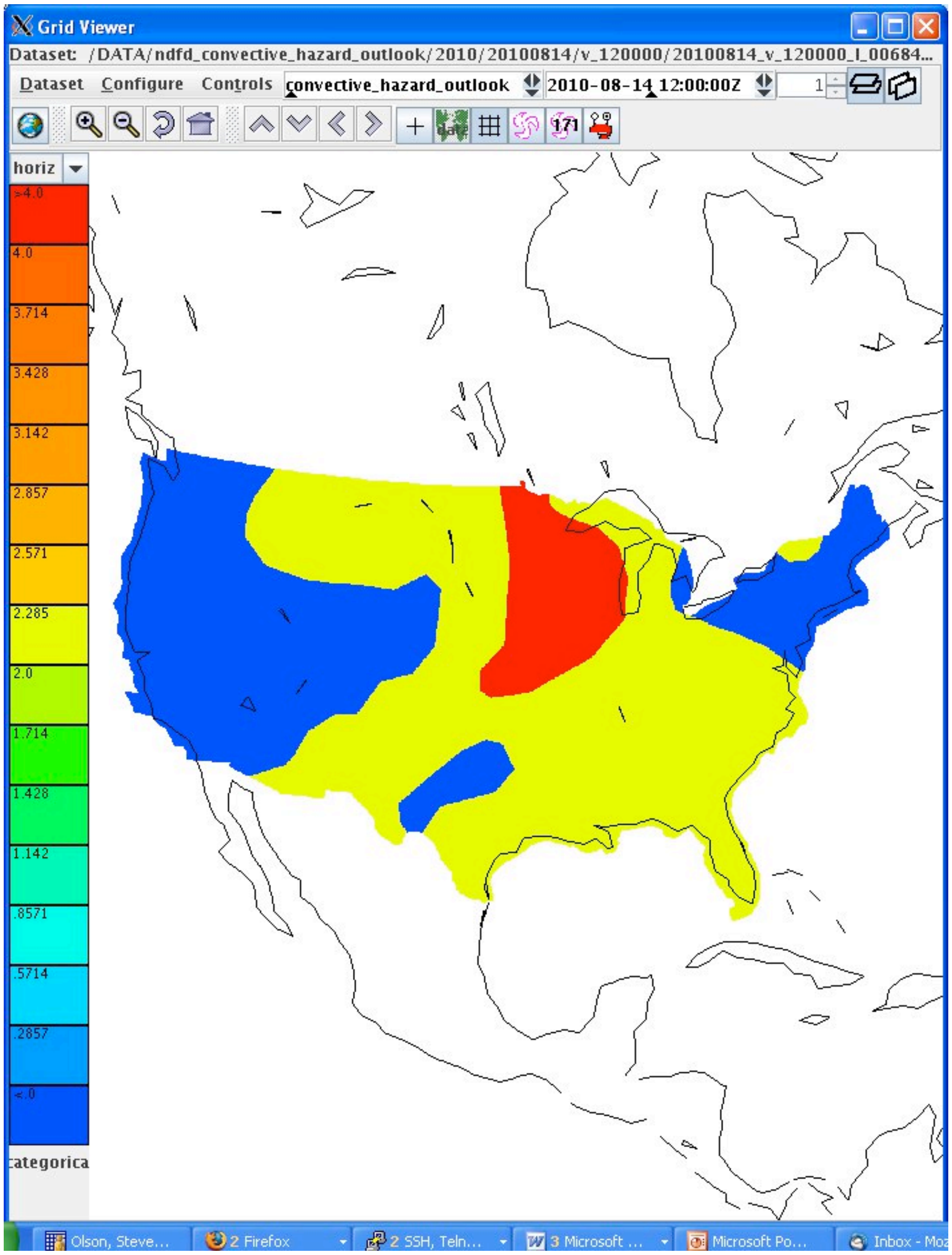

NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump
Grids WMS PointFeature PointObs StationObs Trajectory Images Radial StationRadial FeatureScan
WCS

dataset: /DATA/ndfd_convective_hazard_outlook/2010/20100816/v_120000/20100816_v_120000_l_0241200.nc

coordSystem	description	ens	name	rt	shape	t	units	x	y	z
time y x	Slight risk convec...		convective_hazar...		1,689,1073	time	categorical	x	y	
y x	longitude		longitude		689,1073		degrees_east	x	y	
y x	latitude		latitude		689,1073		degrees_north	x	y	

axisType	coordVar	description	dims	name	positive	regular	shape	units
GeoY	✓	y coordinate of proje...	y	y		5.079405806785406	689	km
GeoX	✓	x coordinate of proje...	x	x		5.07940582730876...	1073	km
Time	✓	time	time	time		0.0	1	hours since 2010-08...

15. Click on the convective hazard outlook tab and then click the red viewer icon. A new GUI window will appear. Initially, the viewer window should be completely black. Clicking on the red icon one more time will display the image. The viewer GUI should look something like:



8.2.4.4 *Test Data Reduction and Analysis*

All data collected automatically from the PC during testing will be archived and stored for future reference. All issues that arise during the test will be documented in a Problem Tracking Report (PTR) summarizing the description, criticality and proposed solution of the issue.

8.3 NDGD GLMP Ceiling Height dataset

8.3.1 Introduction

8.3.1.1 Purpose and Scope

The purpose of this test is to demonstrate the way the 4-D Wx Data Cube may operate with National Digital Forecast Database (NDFD) Wind Speed data at initial operating capacity. The publishing and discovery of metadata about the WCS service and the NDFD Wind Speed dataset itself from federated reg/rep agency repositories will be tested. The requirements of the WCS RI are located in Appendix C of the Capability Evaluation Plan. System level requirements in Appendix D. Latencies, although not required, can be measured.

8.3.2 Reference Documents

1. A NWS NDGD Localized Aviation MOS Program (LAMP) documentation page can be found at http://www.nws.noaa.gov/mdl/gfslamp/docs/gfslamp_info.shtml
2. A description of the LAMP system can be found at http://www.nws.noaa.gov/mdl/lamp/lamp_info.shtml
3. A technical paper describing gridding LAMP guidance for aviation forecasting can be found at: http://ams.confex.com/ams/90annual/techprogram/paper_160491.htm

8.3.3 Test Description

8.3.3.1 System Under Test

MDL will be the source system which will be providing NDFD/NDGD data via NOAAnet

8.3.3.2 Test Setup

Prior to the test, the evaluation PC must be loaded with RedHat 5 O/S and configured with OGC compliant software to request and display weather data from the Cube. Unidata's IDV must also be installed to verify that the dataset is available.

a. Test Equipment at the Tech Center

   NAS Simulator, evaluation PC running the IDV, GSD's Testing Portal

b. Test Equipment at remote site

   MDL's WCSRI endpoint with live data

8.3.3.3 Personnel

FAA WJHTC, MDL, GSD

8.3.3.4 Client Location

NWEC lab at WJHTC

8.3.3.5 Server Location

NOAA/NWS/MDL

8.3.4 Test Conduct

8.3.4.1 Security Concerns

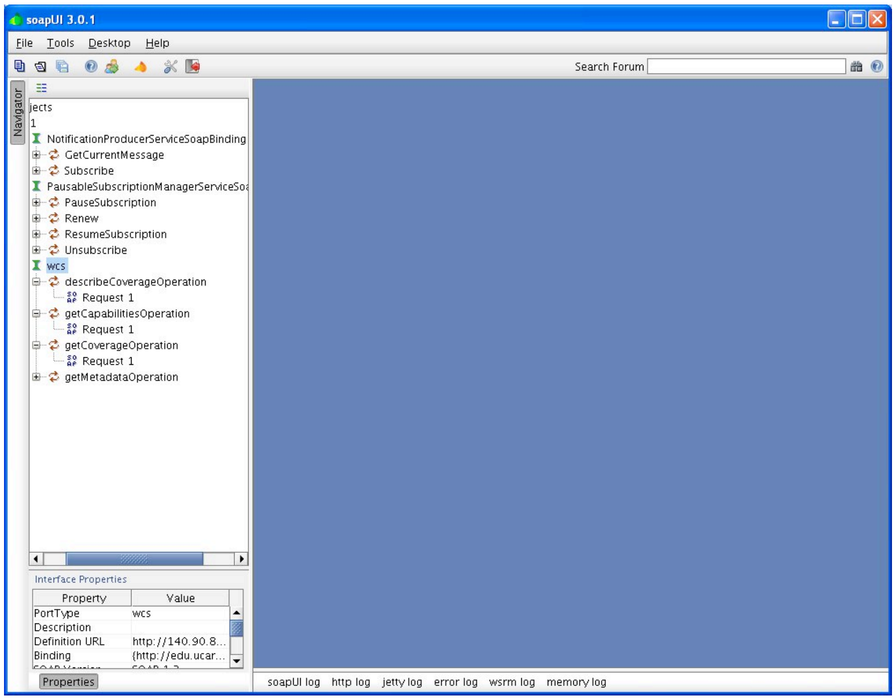
Performance on operational system – to get around this do it at 19Z. Least impact to FTI. Compliant with NIST guided, NOAA/FAA policies on security architecture and relevant information security technologies through a XML gateway.

8.3.4.2 Requirements Under Test

- RI shall support all WCSRI 1.1.0/2.0 spec requirements in Appendix C of the current FY10 Capability Evaluation Plan
- Though not required, the system performance (e.g. latency) shall be measured/baselined.

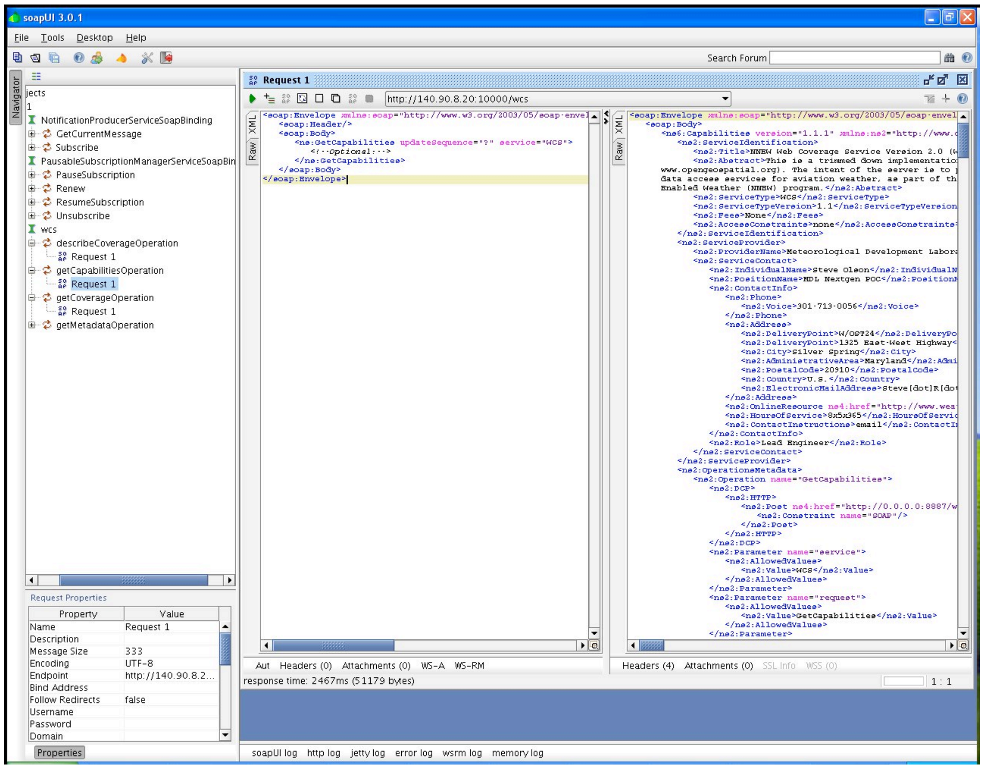
8.3.4.3 Procedures

1. Start up soapUI version 3.0+.
    - Import mdl-wcs-soapui-project.xml file if needed. Click on **File**, then **Import Project,** then select file **mdl-wcs-soapui-project.xml , and click on Open**.
2. Click on the "plus" sign and this will expand the **wcs** to include the following SOAP functions:
    - **describeCoverageOperation**
    - **getCapabilitiesOperation**
    - **getCoverageOperation**
    - **getMetadataOperation**



Property	Value
PortType	wcs
Description	
Definition URL	http://140.90.8...
Binding	{http://edu.ucar...

3. Click the "+" symbol for each of these features, and you will reveal a "SOAP Request 1". These soap requests serve as the interface between the user, and our WCS (where the data resides).
4. Step one: Perform a getCapabilities SOAP request using soapUI:
   - Double click on the "SOAP Request 1" associated with getCapabilitiesOperation. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server. There are no fields that need to be populated with this request. This is a simple request to find out what data is available and more background information about our WCS.
   - Click on the green arrow (Submit request to a specified endpoint URL). After execution, a getCapabilities soap response will appear in the right-hand pane indicating some background information about who's offering the service, and what data is available via our WCS.
   - The most important information that is contained in this soap response is the "**ns2:parameter name="Identifier**". This will provide "Values" indicating what data is available. The "urn: value form this soap response is an important input to the second request.
   - Copy the urn information (**urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDGD-CONUS-Gridded-LAMP-Ceiling-Height**) into the copy and paste buffer.


soapUI 3.0.1
File Tools Desktop Help
Search Forum
Navigator
jects
1
NotificationProducerServiceSoapBinding
GetCurrentMessage
Subscribe
PausableSubscriptionManagerServiceSoapBin
PauseSubscription
Renew
ResumeSubscription
Unsubscribe
wcs
describeCoverageOperation
Request 1
getCapabilitiesOperation
Request 1
getCoverageOperation
Request 1
getMetadataOperation
Request 1
http://140.90.8.20:10000/wcs
XML
Raw
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (0) SSL Info WSS (0)
1 : 1
soapUI log http log jetty log error log wsrm log memory log
Properties

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envel
   <soap:Header/>
   <soap:Body>
      <ns:GetCapabilities updateSequence="?" service="WCS">
         <!--Optional:-->
      </ns:GetCapabilities>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envel
   <soap:Body>
      <ns6:Capabilities version="1.1.1" xmlns:ns2="http://www.o
         <ns2:ServiceIdentification>
            <ns2:Title>NNEW Web Coverage Service Version 2.0 (W
            <ns2:Abstract>This is a trimmed down implementatio
      www.opengeospatial.org). The intent of the server is to p
      data access services for aviation weather, as part of th
      Enabled Weather (NNEW) program.</ns2:Abstract>
            <ns2:ServiceType>WCS</ns2:ServiceType>
            <ns2:ServiceTypeVersion>1.1</ns2:ServiceTypeVersion
            <ns2:Fees>None</ns2:Fees>
            <ns2:AccessConstraints>none</ns2:AccessConstraints
         </ns2:ServiceIdentification>
         <ns2:ServiceProvider>
            <ns2:ProviderName>Meteorological Development Labora
            <ns2:ServiceContact>
               <ns2:IndividualName>Steve Olson</ns2:IndividualN
               <ns2:PositionName>MDL Nextgen POC</ns2:PositionN
               <ns2:ContactInfo>
                  <ns2:Phone>
                     <ns2:Voice>301-713-0056</ns2:Voice>
                  </ns2:Phone>
                  <ns2:Address>
                     <ns2:DeliveryPoint>W/OST24</ns2:DeliveryPo
                     <ns2:DeliveryPoint>1325 East-West Highway<
                     <ns2:City>Silver Spring</ns2:City>
                     <ns2:AdministrativeArea>Maryland</ns2:Admi
                     <ns2:PostalCode>20910</ns2:PostalCode>
                     <ns2:Country>U.S.</ns2:Country>
                     <ns2:ElectronicMailAddress>Steve[dot]R[do
                  </ns2:Address>
                  <ns2:OnlineResource ns4:href="http://www.wea
                  <ns2:HoursOfService>8x5x365</ns2:HoursOfServic
                  <ns2:ContactInstructions>email</ns2:ContactIn
               </ns2:ContactInfo>
               <ns2:Role>Lead Engineer</ns2:Role>
            </ns2:ServiceContact>
         </ns2:ServiceProvider>
         <ns2:OperationsMetadata>
            <ns2:Operation name="GetCapabilities">
               <ns2:DCP>
                  <ns2:HTTP>
                     <ns2:Post ns4:href="http://0.0.0.0:8887/w
                        <ns2:Constraint name="SOAP"/>
                     </ns2:Post>
                  </ns2:HTTP>
               </ns2:DCP>
               <ns2:Parameter name="service">
                  <ns2:AllowedValues>
                     <ns2:Value>WCS</ns2:Value>
                  </ns2:AllowedValues>
               </ns2:Parameter>
               <ns2:Parameter name="request">
                  <ns2:AllowedValues>
                     <ns2:Value>GetCapabilities</ns2:Value>
                  </ns2:AllowedValues>
               </ns2:Parameter>
```

response time: 2467ms (51179 bytes)

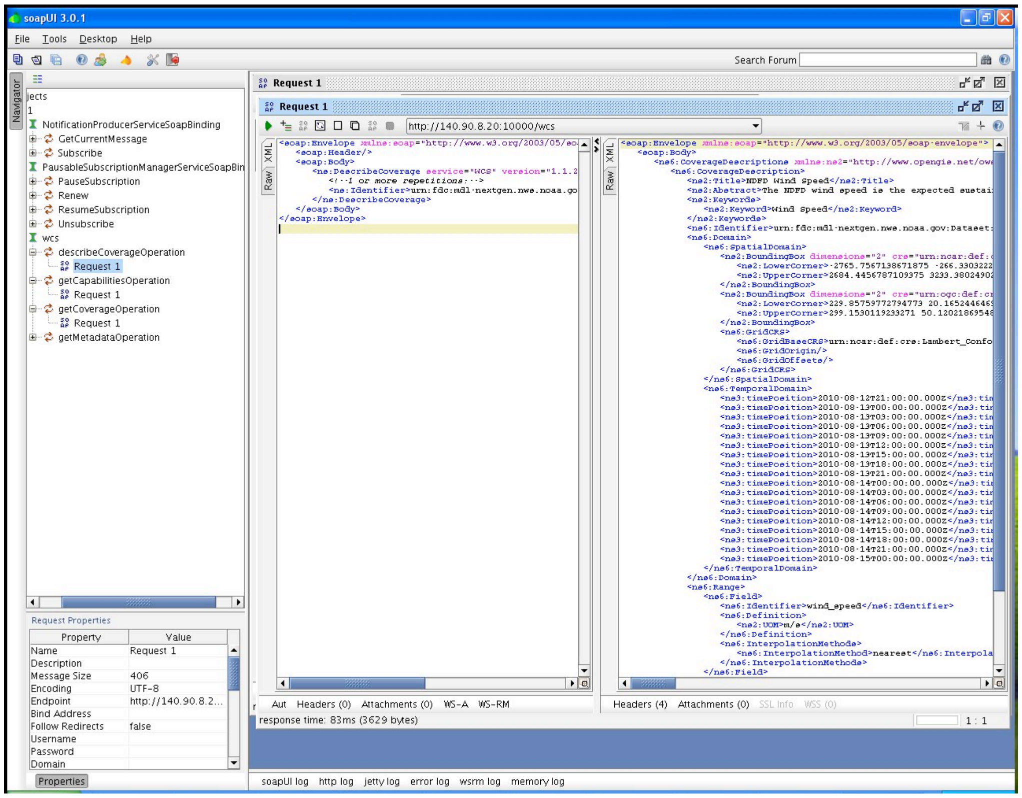
Request Properties

Property	Value
Name	Request 1
Description	
Message Size	333
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

5. Step two: Perform a describeCoverage SOAP request using soapUI:
   - Double click on the “SOAP Request 1” associated with the describeCoverageOption. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server.
   - Cut and paste the following information in the left-hand pane, and click on green arrow, or replace the “**?**” with “**urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDGD-CONUS-Gridded-LAMP-Ceiling-Height**”:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-
envelope" xmlns:ns="http://www.opengis.net/wcs/1.1">
   <soap:Header/>
   <soap:Body>
      <ns:DescribeCoverage service="WCS" version="1.1.2">
         <!--1 or more repetitions:-->
   <ns:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDGD-
CONUS-Gridded-LAMP-Ceiling-Height</ns:Identifier>
      </ns:DescribeCoverage>
   </soap:Body>
</soap:Envelope>
```

- The screen should look something like this:


soapUI 3.0.1
File Tools Desktop Help
Search Forum
Navigator
jects
1
NotificationProducerServiceSoapBinding
GetCurrentMessage
Subscribe
PausableSubscriptionManagerServiceSoapBin
PauseSubscription
Renew
ResumeSubscription
Unsubscribe
wcs
describeCoverageOperation
Request 1
getCapabilitiesOperation
Request 1
getCoverageOperation
Request 1
getMetadataOperation
Request 1
Request 1
http://140.90.8.20:10000/wcs
XML
Raw
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (0) SSL Info WSS (0)
1 : 1
soapUI log http log jetty log error log wsrm log memory log
Properties

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/so
   <soap:Header/>
   <soap:Body>
      <ns:DescribeCoverage service="WCS" version="1.1.2
         <!--1 or more repetitions:-->
         <ns:Identifier>urn:fdc:mdl-nextgen.nws.noaa.go
      </ns:DescribeCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Body>
      <ns6:CoverageDescriptions xmlns:ns2="http://www.opengis.net/ows
         <ns6:CoverageDescription>
            <ns2:Title>NDFD Wind Speed</ns2:Title>
            <ns2:Abstract>The NDFD wind speed is the expected sustai
            <ns2:Keywords>
               <ns2:Keyword>Wind Speed</ns2:Keyword>
            </ns2:Keywords>
            <ns6:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:
            <ns6:Domain>
               <ns6:SpatialDomain>
                  <ns2:BoundingBox dimensions="2" crs="urn:ncar:def:
                     <ns2:LowerCorner>-2765.7567138671875 -266.3303222
                     <ns2:UpperCorner>2684.4456787109375 3233.38024902
                  </ns2:BoundingBox>
                  <ns2:BoundingBox dimensions="2" crs="urn:ogc:def:cr
                     <ns2:LowerCorner>229.85759772794773 20.165244646
                     <ns2:UpperCorner>299.1530119233271 50.1202186954
                  </ns2:BoundingBox>
                  <ns6:GridCRS>
                     <ns6:GridBaseCRS>urn:ncar:def:crs:Lambert_Confo
                     <ns6:GridOrigin/>
                     <ns6:GridOffsets/>
                  </ns6:GridCRS>
               </ns6:SpatialDomain>
               <ns6:TemporalDomain>
                  <ns3:timePosition>2010-08-12T21:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T00:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T03:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T06:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T09:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T12:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T15:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T18:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T21:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T00:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T03:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T06:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T09:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T12:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T15:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T18:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T21:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-15T00:00:00.000Z</ns3:tim
               </ns6:TemporalDomain>
            </ns6:Domain>
            <ns6:Range>
               <ns6:Field>
                  <ns6:Identifier>wind_speed</ns6:Identifier>
                  <ns6:Definition>
                     <ns2:UOM>m/s</ns2:UOM>
                  </ns6:Definition>
                  <ns6:InterpolationMethods>
                     <ns6:InterpolationMethod>nearest</ns6:Interpola
                  </ns6:InterpolationMethods>
               </ns6:Field>
```

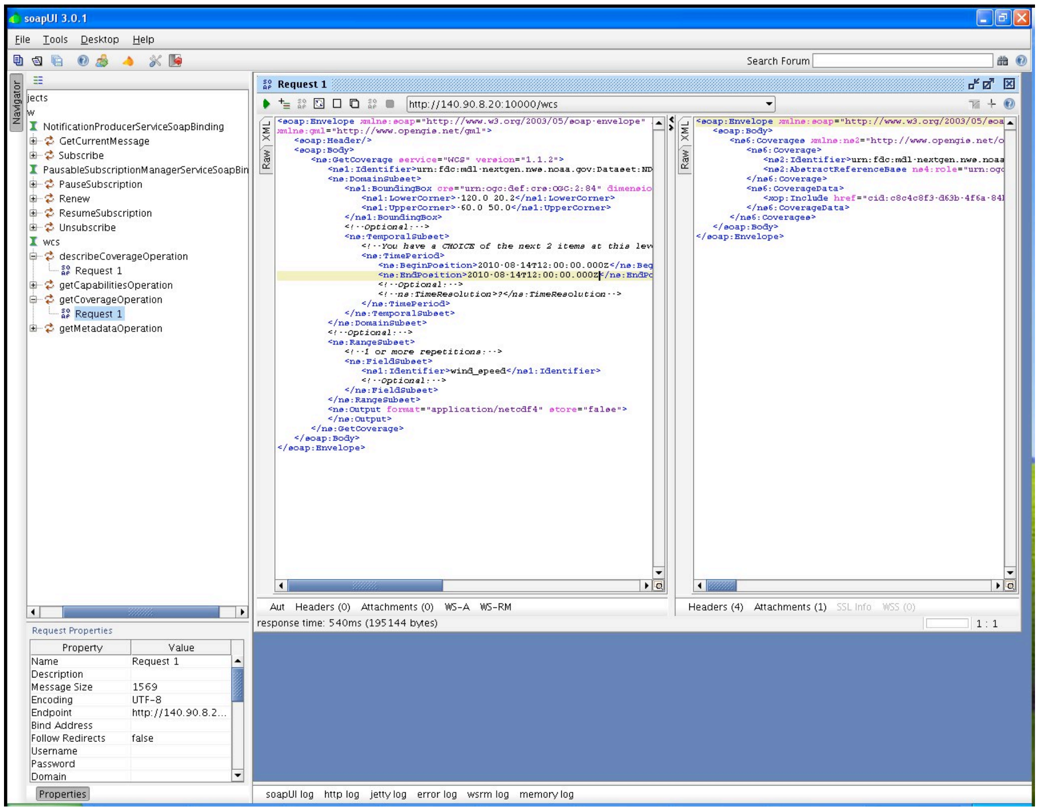
response time: 83ms (3629 bytes)

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	406
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

6. Step three: Perform a getCoverage SOAP request using soapUI:
   - Double click on the “SOAP Request 1” associated with the getCoverageOption. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server.
   - Next, we will need to populate the correct getCoverage information in the left-side window pane. The easiest way to accomplish this is through a simple cut and paste operation. Cut and paste the following information in the left-hand pane. There is one field that will need to be adjusted, and that is the beginPosition and endPosition times. From the describeCoverage info, select the starting (beginPosition) and ending (endPosition) times you’d like to receive data for, the click on green arrow:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns="http://www.opengis.net/wcs/1.1"
xmlns:ns1="http://www.opengis.net/ows/1.1"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDGD-
CONUS-Gridded-LAMP-Ceiling-Height</ns1:Identifier>
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84"
dimensions="2">
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this level-
->
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-
04T00:00:00.000Z</ns:BeginPosition>
                  <ns:EndPosition>2010-08-
04T00:00:00.000Z</ns:EndPosition>
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>
               <ns1:Identifier>ceiling</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```



http://140.90.8.20:10000/wcs

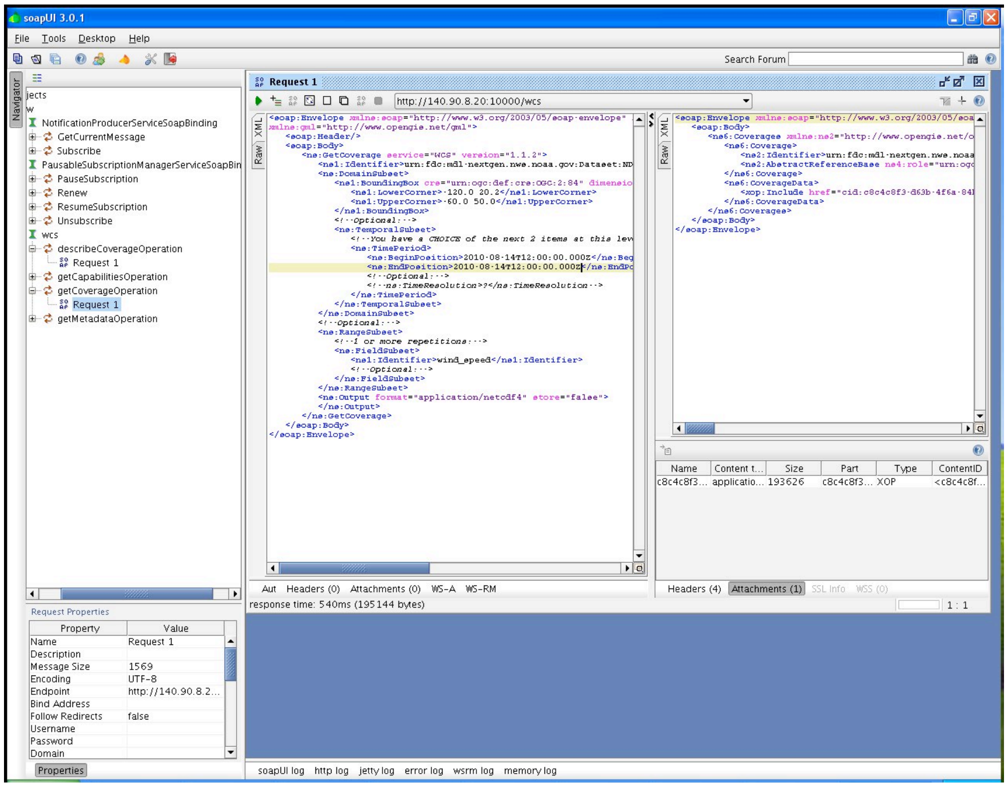
```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:ND
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84" dimensio
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this lev
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-14T12:00:00.000Z</ns:Beg
                  <ns:EndPosition>2010-08-14T12:00:00.000Z</ns:EndPo
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>
               <ns1:Identifier>wind_speed</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soa
   <soap:Body>
      <ns6:Coverages xmlns:ns2="http://www.opengis.net/c
         <ns6:Coverage>
            <ns2:Identifier>urn:fdc:mdl-nextgen.nws.noaa
            <ns2:AbstractReferenceBase ns4:role="urn:ogc
         </ns6:Coverage>
         <ns6:CoverageData>
            <xop:Include href="cid:c8c4c8f3-d63b-4f6a-84
         </ns6:CoverageData>
      </ns6:Coverages>
   </soap:Body>
</soap:Envelope>
```

response time: 540ms (195144 bytes)

Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

7. After execution, a getCoverage soap response will appear in the right-hand pane showing whether or not there was a successful request. If successful, there should be an attachment containing a netCDF4 file. Look for the word “Attachments” towards to lower left-hand corner of the getCoverage soap response (right-side pane).


soapUI 3.0.1
File Tools Desktop Help
Search Forum
Navigator
jects
NotificationProducerServiceSoapBinding
GetCurrentMessage
Subscribe
PausableSubscriptionManagerServiceSoapBin
PauseSubscription
Renew
ResumeSubscription
Unsubscribe
wcs
describeCoverageOperation
Request 1
getCapabilitiesOperation
getCoverageOperation
Request 1
getMetadataOperation
Request 1
http://140.90.8.20:10000/wcs
XML
Raw
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (1) SSL Info WSS (0)
Properties
soapUI log http log jetty log error log wsrm log memory log

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:ND
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84" dimensio
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this lev
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-14T12:00:00.000Z</ns:Beg
                  <ns:EndPosition>2010-08-14T12:00:00.000Z</ns:EndPo
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>
               <ns1:Identifier>wind_speed</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soa
   <soap:Body>
      <ns6:Coverages xmlns:ns2="http://www.opengis.net/o
         <ns6:Coverage>
            <ns2:Identifier>urn:fdc:mdl-nextgen.nws.noaa
            <ns2:AbstractReferenceBase ns4:role="urn:ogc
         </ns6:Coverage>
         <ns6:CoverageData>
            <xop:Include href="cid:c8c4c8f3-d63b-4f6a-84
         </ns6:CoverageData>
      </ns6:Coverages>
   </soap:Body>
</soap:Envelope>
```

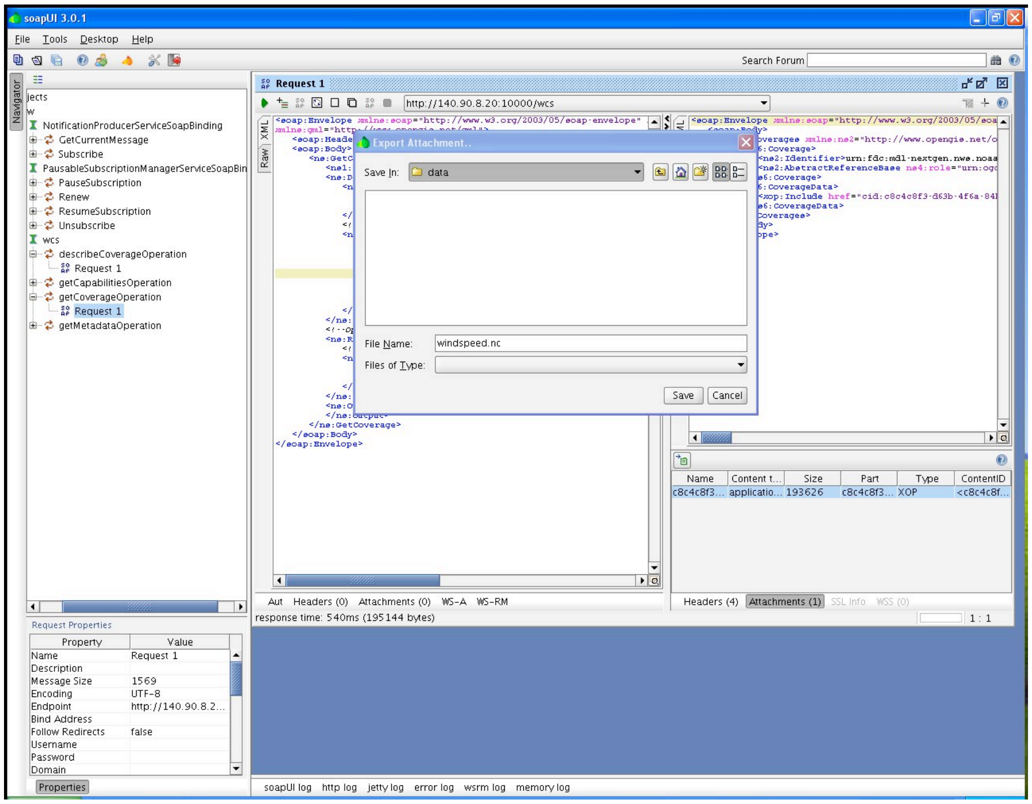
Name	Content t...	Size	Part	Type	ContentID
c8c4c8f3...	applicatio...	193626	c8c4c8f3...	XOP	<c8c4c8f...

response time: 540ms (195144 bytes)

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

8. Click once on the word “Attachments”. A panel with “Name”, “Content type”, etc will appear. Below that line is the netCDF4 file.
9. Click once anywhere on this line. A green tab will appear that allows for the export of the selected attachment to a file. Click that tab once, and an “Export Attachment” GUI will appear. Save the file to a desired location. A number of programs can be used to analyze the netCDF4 file, including ncview and ncdump.


soapUI 3.0.1
File Tools Desktop Help
Search Forum
Navigator
Request 1
XML
Raw
Export Attachment...
Save In: data
Save
Cancel
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (1) SSL Info WSS (0)
soapUI log http log jetty log error log wsrm log memory log
Properties

```
NotificationProducerServiceSoapBinding
  GetCurrentMessage
  Subscribe
PausableSubscriptionManagerServiceSoapBin
  PauseSubscription
  Renew
  ResumeSubscription
  Unsubscribe
wcs
  describeCoverageOperation
    Request 1
  getCapabilitiesOperation
  getCoverageOperation
    Request 1
  getMetadataOperation
```

http://140.90.8.20:10000/wcs

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
...
      </ne:GetCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soa
   <soap:Body>
overages xmlns:ns2="http://www.opengis.net/o
6:Coverage>
<ns2:Identifier>urn:fdc:mdl-nextgen.nws.noaa
<ns2:AbstractReferenceBase ns4:role="urn:ogc
s6:Coverage>
6:CoverageData>
<xop:Include href="cid:c8c4c8f3-d63b-4f6a-84
s6:CoverageData>
Coverages>
dy>
ope>
```

File Name: windspeed.nc

Files of Type:

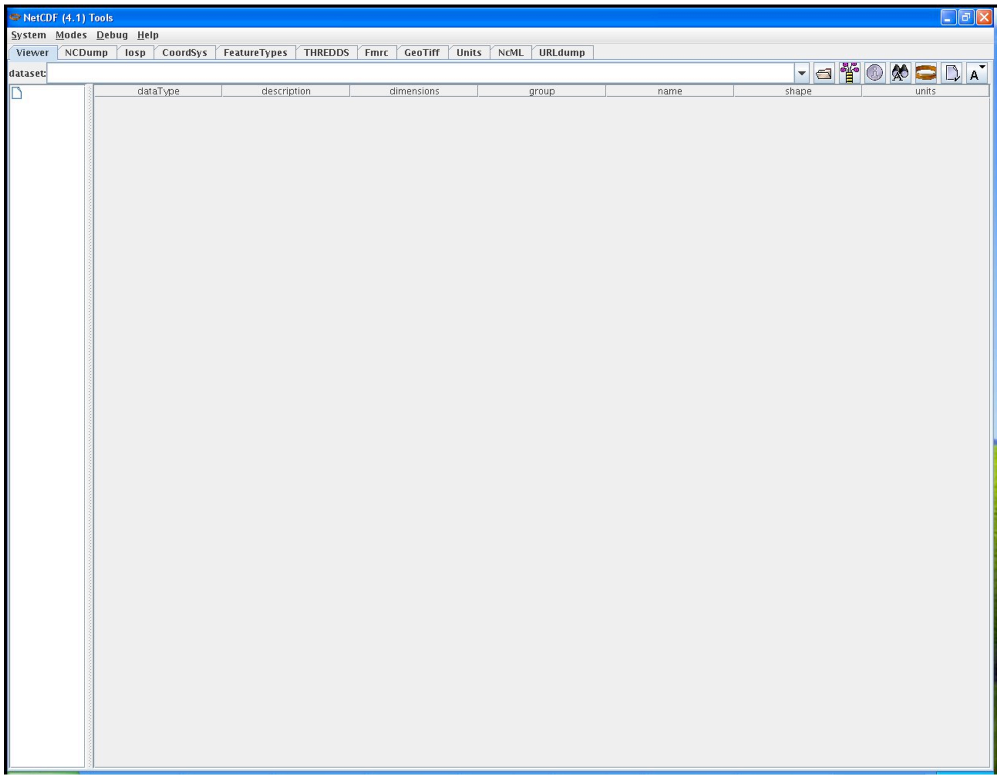
Name	Content t...	Size	Part	Type	ContentID
c8c4c8f3...	applicatio...	193626	c8c4c8f3...	XOP	<c8c4c8f...

response time: 540ms (195144 bytes)

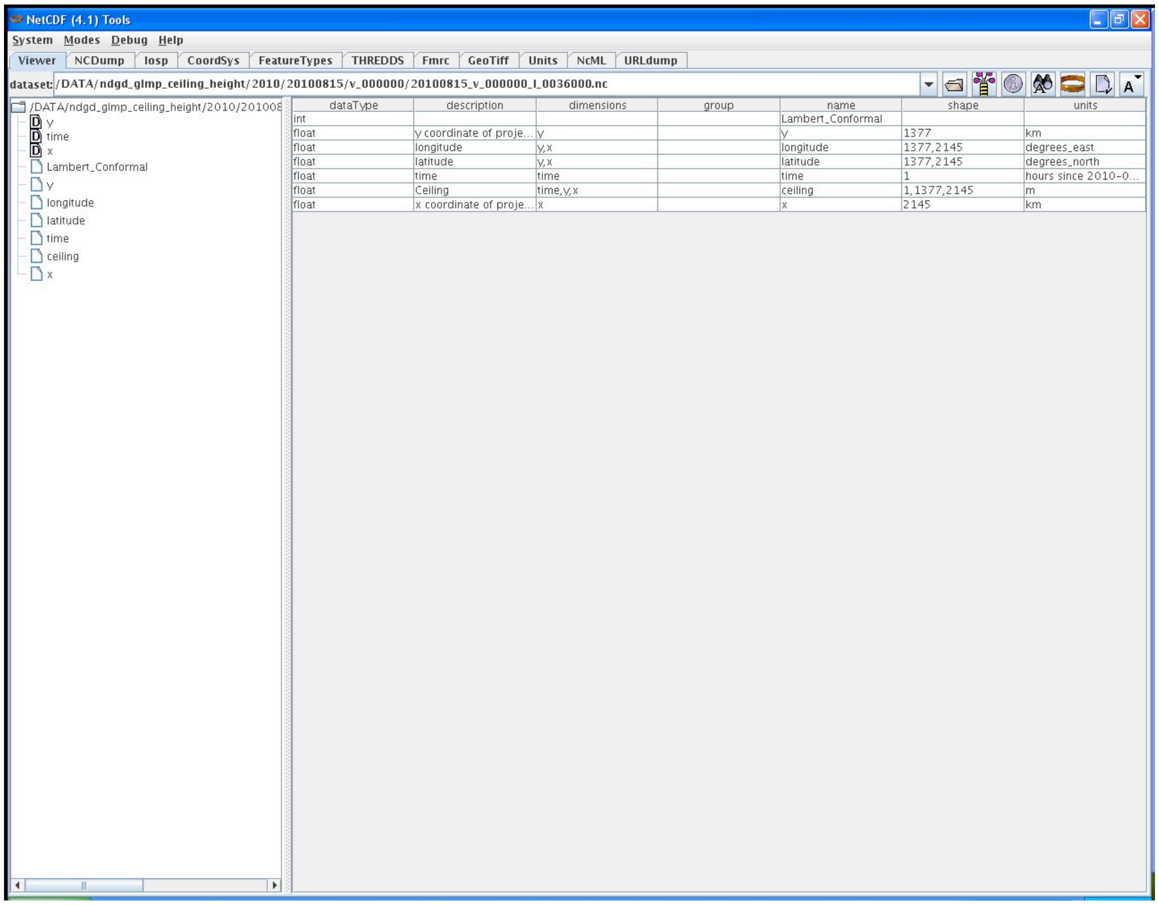
Request Properties

Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

10. A number of programs can be used to view the netCDF4 file, including IDV, toolsUI, ncview and ncdump. We have been using toolsUI version 4.1 (i.e. toolsUI-4.1.jar)
11. Start up toolsUI 4.1. The interface will look something like:


NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump
dataset:
dataType description dimensions group name shape units

12. Click on the open folder icon. Select the image you want to view. Click open, and the toolsUI GUI will look something like:

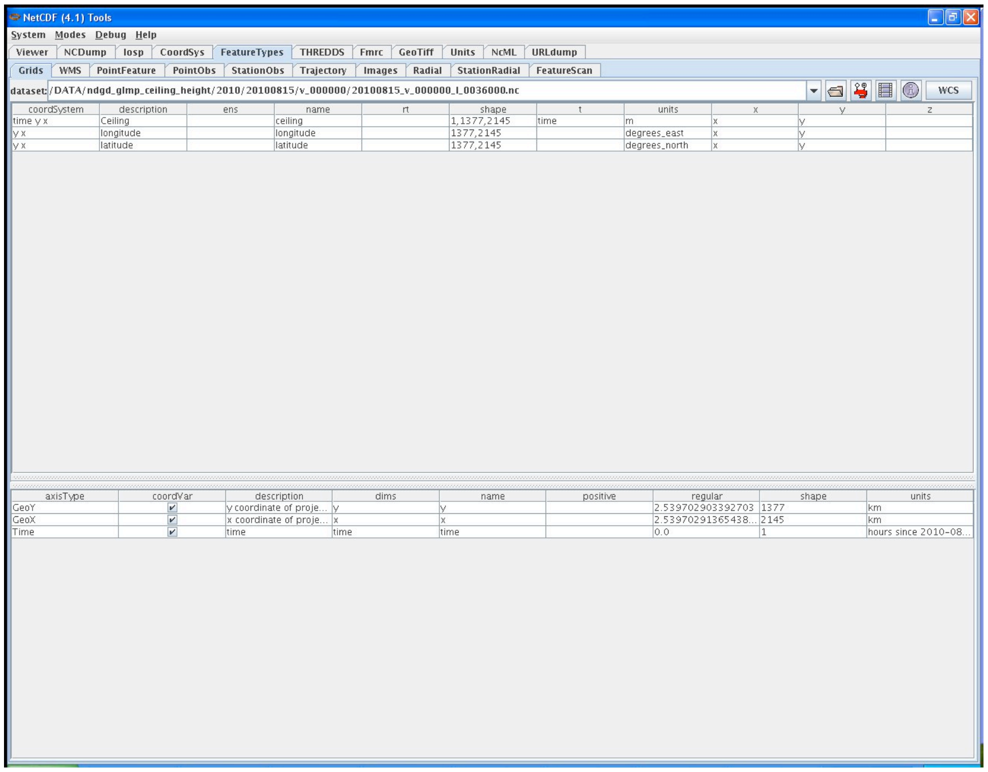

NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump

dataset: /DATA/ndgd_glmp_ceiling_height/2010/20100815/v_000000/20100815_v_000000_l_0036000.nc

```
/DATA/ndgd_glmp_ceiling_height/2010/201008
  D y
  D time
  D x
    Lambert_Conformal
    y
    longitude
    latitude
    time
    ceiling
    x
```

dataType	description	dimensions	group	name	shape	units
int				Lambert_Conformal		
float	y coordinate of proje...	y		y	1377	km
float	longitude	y,x		longitude	1377,2145	degrees_east
float	latitude	y,x		latitude	1377,2145	degrees_north
float	time	time		time	1	hours since 2010-0...
float	Ceiling	time,y,x		ceiling	1,1377,2145	m
float	x coordinate of proje...	x		x	2145	km

13. The viewer tab will describe information about the file. In this case, its NDGD Gridded LAMP Ceiling Height.
14. To view the image, click on the FeatureTypes tab, and click on the open folder icon again. Select the image you want to view (20100815_v_120000_0079200.nc in our example) and click open. The toolsUI GUI will look something like:

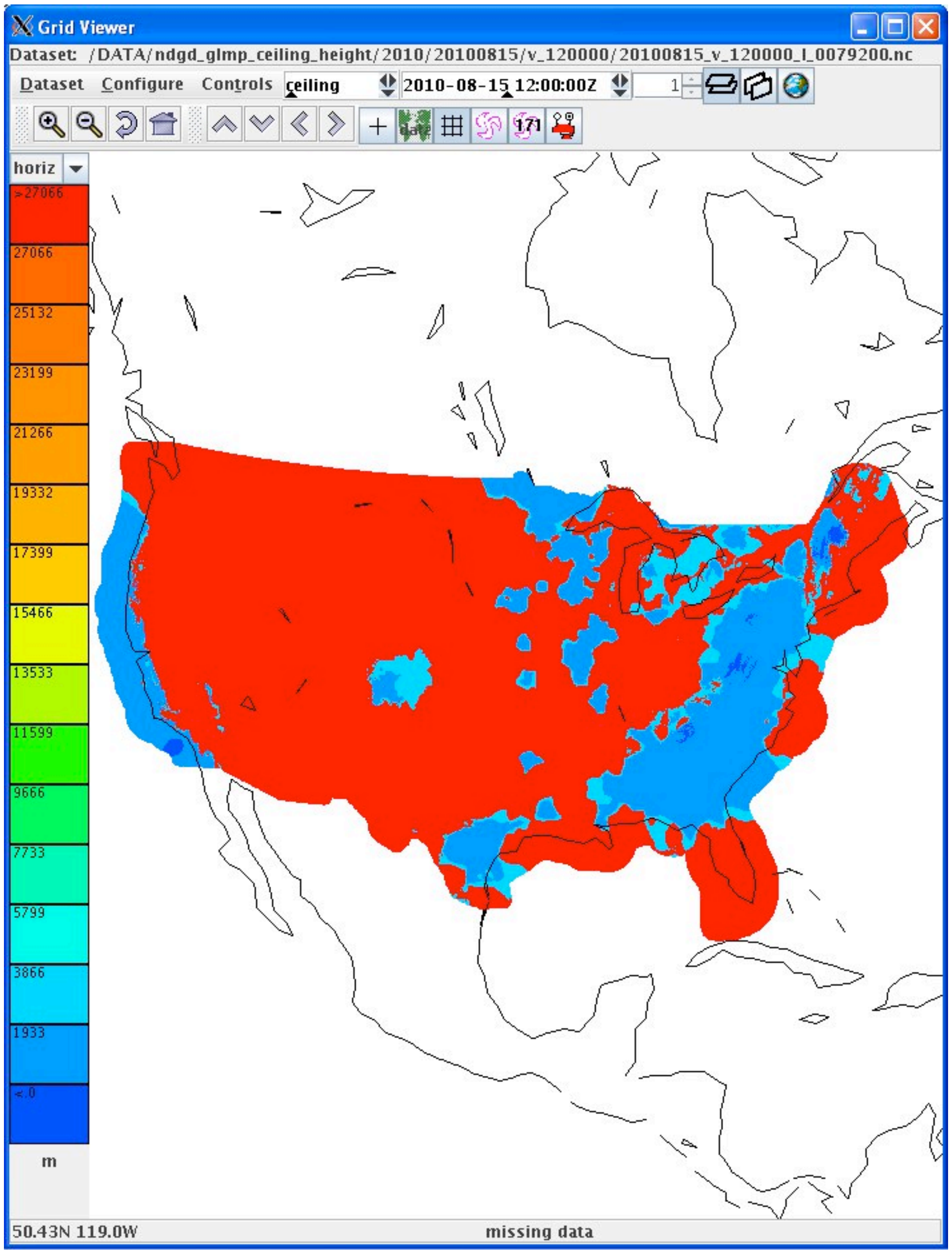

NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump
Grids WMS PointFeature PointObs StationObs Trajectory Images Radial StationRadial FeatureScan
WCS

dataset: /DATA/ndgd_glmp_ceiling_height/2010/20100815/v_000000/20100815_v_000000_l_0036000.nc

coordSystem	description	ens	name	rt	shape	t	units	x	y	z
time y x	Ceiling		ceiling		1,1377,2145	time	m	x	y	
y x	longitude		longitude		1377,2145		degrees_east	x	y	
y x	latitude		latitude		1377,2145		degrees_north	x	y	

axisType	coordVar	description	dims	name	positive	regular	shape	units
GeoY	✓	y coordinate of proje...	y	y		2.539702903392703	1377	km
GeoX	✓	x coordinate of proje...	x	x		2.53970291365438...	2145	km
Time	✓	time	time	time		0.0	1	hours since 2010-08...

15. Click on the ceiling height tab and then click the red viewer icon. A new GUI window will appear. Initially, the viewer window should be completely black. Clicking on the red icon one more time will display the image. The viewer GUI should look something like:


Dataset: /DATA/ndgd_glmp_ceiling_height/2010/20100815/v_120000/20100815_v_120000_l_0079200.nc

ceiling 2010-08-15 12:00:00Z

50.43N 119.0W missing data

8.3.4.4 *Test Data Reduction and Analysis*

All data collected automatically from the PC during testing will be archived and stored for future reference. All issues that arise during the test will be documented in a Problem Tracking Report (PTR) summarizing the description, criticality and proposed solution of the issue.

8.4 NDGD GLMP Visibility dataset

8.4.1 Introduction

8.4.1.1 Purpose and Scope

The purpose of this test is to demonstrate the way the 4-D Wx Data Cube may operate with National Digital Forecast Database (NDFD) Wind Speed data at initial operating capacity. The publishing and discovery of metadata about the WCS service and the NDFD Wind Speed dataset itself from federated reg/rep agency repositories will be tested. The requirements of the WCS RI are located in Appendix C of the Capability Evaluation Plan. System level requirements in Appendix D. Latencies, although not required, can be measured.

8.4.2 Reference Documents

1. A NWS NDGD Localized Aviation MOS Program (LAMP) documentation page can be found at http://www.nws.noaa.gov/mdl/gfslamp/docs/gfslamp_info.shtml
2. A description of the LAMP system can be found at http://www.nws.noaa.gov/mdl/lamp/lamp_info.shtml
3. A technical paper describing gridding LAMP guidance for aviation forecasting can be found at: http://ams.confex.com/ams/90annual/techprogram/paper_160491.htm

8.4.3 Test Description

8.4.3.1 System Under Test

MDL will be the source system which will be providing NDFD/NDGD data via NOAAnet

8.4.3.2 Test Setup

Prior to the test, the evaluation PC must be loaded with RedHat 5 O/S and configured with OGC compliant software to request and display weather data from the Cube. Unidata's IDV must also be installed to verify that the dataset is available.

a. Test Equipment at the Tech Center
    NAS Simulator, evaluation PC running the IDV, GSD's Testing Portal
b. Test Equipment at remote site
    MDL's WCSRI endpoint with live data

8.4.3.3 Personnel

FAA WJHTC, MDL, GSD

8.4.3.4 Client Location

NWEC lab at WJHTC

8.4.3.5 Server Location

NOAA/NWS/MDL

8.4.4 Test Conduct

8.4.4.1 Security Concerns

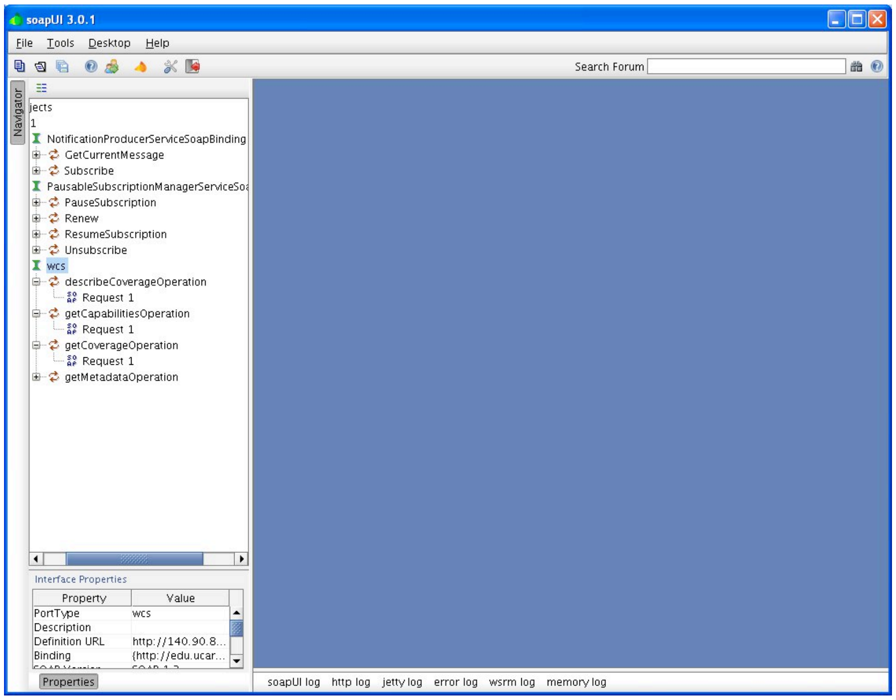
Performance on operational system – to get around this do it at 19Z. Least impact to FTI. Compliant with NIST guided, NOAA/FAA policies on security architecture and relevant information security technologies through a XML gateway.

8.4.4.2 Requirements Under Test

- RI shall support all WCSRI 1.1.0/2.0 spec requirements in Appendix C of the current FY10 Capability Evaluation Plan
- Though not required, the system performance (e.g. latency) shall be measured/baselined.

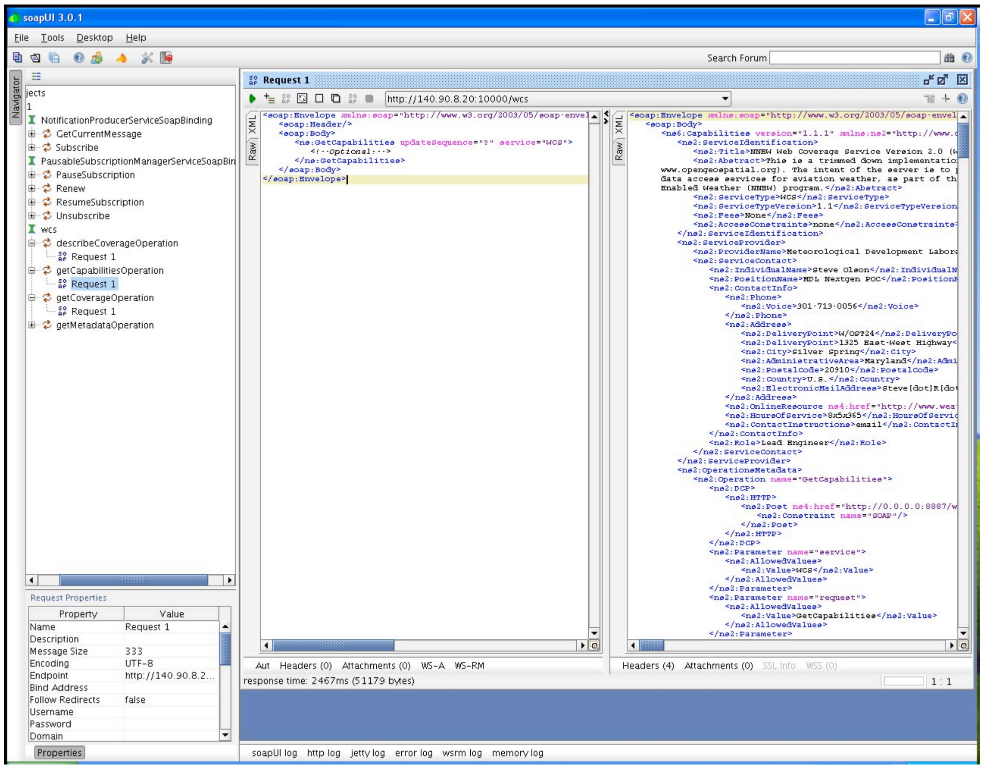
8.4.4.3 Procedures

1. Start up soapUI version 3.0+.
   - Import mdl-wcs-soapui-project.xml file if needed. Click on **File**, then **Import Project,** then select file **mdl-wcs-soapui-project.xml , and click on Open**.
2. Click on the “plus” sign and this will expand the **wcs** to include the following SOAP functions:
   - **describeCoverageOperation**
   - **getCapabilitiesOperation**
   - **getCoverageOperation**
   - **getMetadataOperation**



Property	Value
PortType	wcs
Description	
Definition URL	http://140.90.8...
Binding	{http://edu.ucar...

3. Click the ”+” symbol for each of these features, and you will reveal a “SOAP Request 1”. These soap requests serve as the interface between the user, and our WCS (where the data resides).
4. Step one: Perform a getcapabilities SOAP request using soapUI:
   - Double click on the “SOAP Request 1” associated with getCapabilitiesOperation. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server. There are no fields that need to be populated with this request. This is a simple request to find out what data is available and more background information about our WCS.
   - Click on the green arrow (Submit request to a specified endpoint URL). After execution, a getCapabilities soap response will appear in the right-hand pane indicating some background information about who's offering the service, and what data is available via our WCS.
   - The most important information that is contained in this soap response is the “**ns2:parameter name=“Identifier**”. This will provide “Values” indicating what data is available. The “urn: value form this soap response is an important input to the second request.
   - Copy the urn information (**urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDGD-CONUS-Gridded-LAMP-Visibility**) into the copy and paste buffer.


soapUI 3.0.1
File Tools Desktop Help
Search Forum
Navigator
Request 1
XML
Raw
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (0) SSL Info WSS (0)
1 : 1
Properties
soapUI log http log jetty log error log wsrm log memory log

```
jects
1
NotificationProducerServiceSoapBinding
  GetCurrentMessage
  Subscribe
PausableSubscriptionManagerServiceSoapBin
  PauseSubscription
  Renew
  ResumeSubscription
  Unsubscribe
wcs
  describeCoverageOperation
    Request 1
  getCapabilitiesOperation
    Request 1
  getCoverageOperation
    Request 1
  getMetadataOperation
```

http://140.90.8.20:10000/wcs

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envel
   <soap:Header/>
   <soap:Body>
      <ns:GetCapabilities updateSequence="?" service="WCS">
         <!--Optional:-->
      </ns:GetCapabilities>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envel
   <soap:Body>
      <ns6:Capabilities version="1.1.1" xmlns:ns2="http://www.c
         <ns2:ServiceIdentification>
            <ns2:Title>NNEW Web Coverage Service Version 2.0 (W
            <ns2:Abstract>This is a trimmed down implementatio
         www.opengeospatial.org). The intent of the server is to p
         data access services for aviation weather, as part of th
         Enabled Weather (NNEW) program.</ns2:Abstract>
            <ns2:ServiceType>WCS</ns2:ServiceType>
            <ns2:ServiceTypeVersion>1.1</ns2:ServiceTypeVersion
            <ns2:Fees>None</ns2:Fees>
            <ns2:AccessConstraints>none</ns2:AccessConstraints
         </ns2:ServiceIdentification>
         <ns2:ServiceProvider>
            <ns2:ProviderName>Meteorological Development Labora
            <ns2:ServiceContact>
               <ns2:IndividualName>Steve Olson</ns2:IndividualN
               <ns2:PositionName>MDL Nextgen POC</ns2:PositionN
               <ns2:ContactInfo>
                  <ns2:Phone>
                     <ns2:Voice>301-713-0056</ns2:Voice>
                  </ns2:Phone>
                  <ns2:Address>
                     <ns2:DeliveryPoint>W/OST24</ns2:DeliveryPo
                     <ns2:DeliveryPoint>1325 East-West Highway<
                     <ns2:City>Silver Spring</ns2:City>
                     <ns2:AdministrativeArea>Maryland</ns2:Admi
                     <ns2:PostalCode>20910</ns2:PostalCode>
                     <ns2:Country>U.S.</ns2:Country>
                     <ns2:ElectronicMailAddress>Steve[dot]R[do
                  </ns2:Address>
                  <ns2:OnlineResource ns4:href="http://www.wea
                  <ns2:HoursOfService>8x5x365</ns2:HoursOfServic
                  <ns2:ContactInstructions>email</ns2:ContactIn
               </ns2:ContactInfo>
               <ns2:Role>Lead Engineer</ns2:Role>
            </ns2:ServiceContact>
         </ns2:ServiceProvider>
         <ns2:OperationsMetadata>
            <ns2:Operation name="GetCapabilities">
               <ns2:DCP>
                  <ns2:HTTP>
                     <ns2:Post ns4:href="http://0.0.0.0:8887/w
                        <ns2:Constraint name="SOAP"/>
                     </ns2:Post>
                  </ns2:HTTP>
               </ns2:DCP>
               <ns2:Parameter name="service">
                  <ns2:AllowedValues>
                     <ns2:Value>WCS</ns2:Value>
                  </ns2:AllowedValues>
               </ns2:Parameter>
               <ns2:Parameter name="request">
                  <ns2:AllowedValues>
                     <ns2:Value>GetCapabilities</ns2:Value>
                  </ns2:AllowedValues>
               </ns2:Parameter>
```

Request Properties

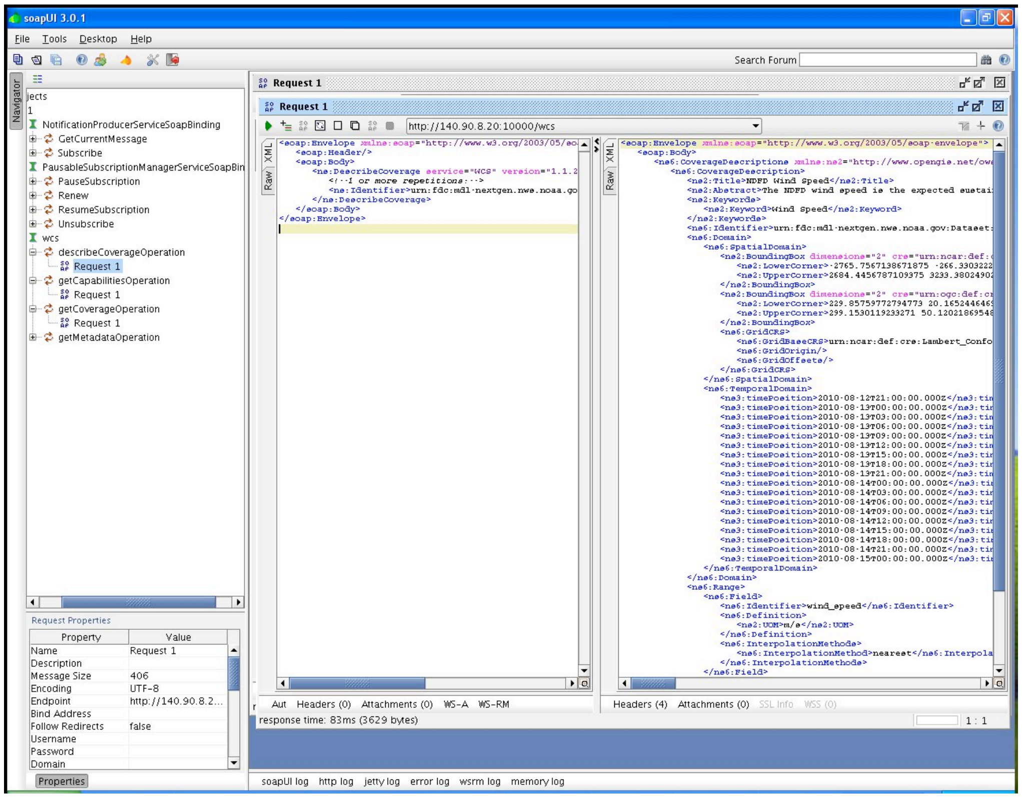
Property	Value
Name	Request 1
Description	
Message Size	333
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

response time: 2467ms (51179 bytes)

5. Step two: Perform a describeCoverage SOAP request using soapUI:
   - Double click on the "SOAP Request 1" associated with the describeCoverageOption. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server.
   - Cut and paste the following information in the left-hand pane, and click on green arrow, or replace the "**?**" with "**urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDGD-CONUS-Gridded-LAMP-Visibility**":

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-
envelope" xmlns:ns="http://www.opengis.net/wcs/1.1">
   <soap:Header/>
   <soap:Body>
      <ns:DescribeCoverage service="WCS" version="1.1.2">
         <!--1 or more repetitions:-->
   <ns:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDGD-
CONUS-Gridded-LAMP-Visibility</ns:Identifier>
      </ns:DescribeCoverage>
   </soap:Body>
</soap:Envelope>
```

- The screen should look something like this:



Property	Value
Name	Request 1
Description	
Message Size	406
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

http://140.90.8.20:10000/wcs

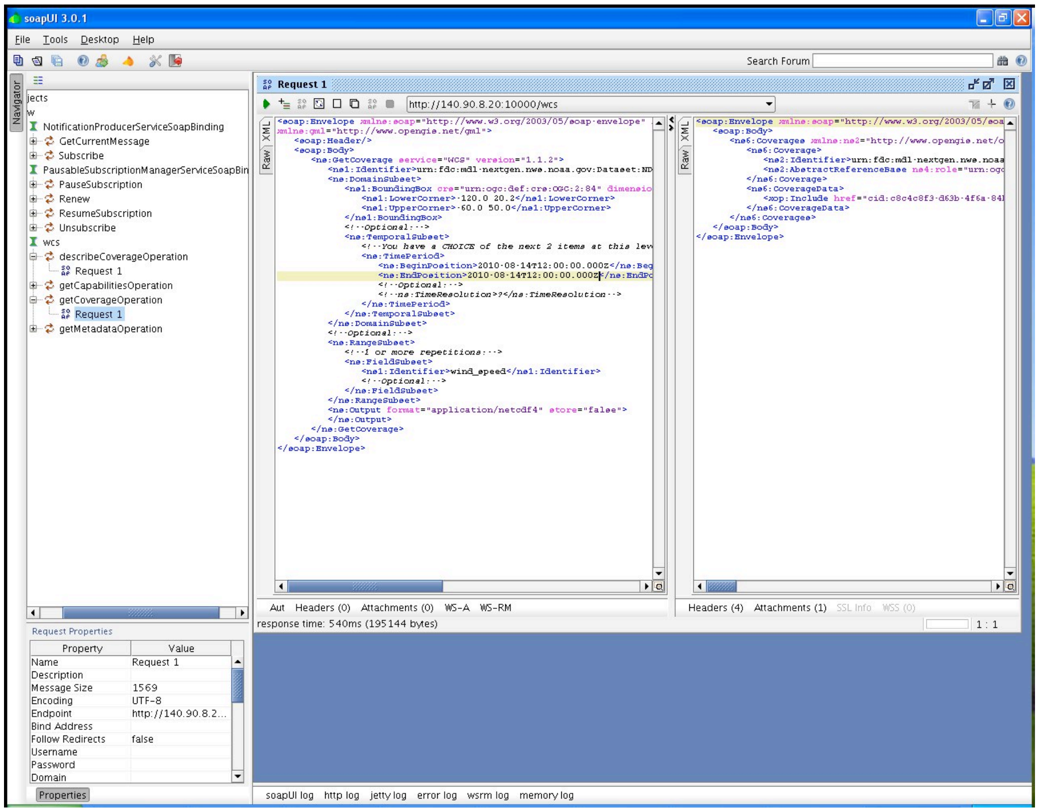
```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/so
   <soap:Header/>
   <soap:Body>
      <ns:DescribeCoverage service="WCS" version="1.1.2
         <!--1 or more repetitions:-->
         <ns:Identifier>urn:fdc:mdl-nextgen.nws.noaa.go
      </ns:DescribeCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Body>
      <ns6:CoverageDescriptions xmlns:ns2="http://www.opengis.net/ows
         <ns6:CoverageDescription>
            <ns2:Title>NDFD Wind Speed</ns2:Title>
            <ns2:Abstract>The NDFD wind speed is the expected sustai
            <ns2:Keywords>
               <ns2:Keyword>Wind Speed</ns2:Keyword>
            </ns2:Keywords>
            <ns6:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:
            <ns6:Domain>
               <ns6:SpatialDomain>
                  <ns2:BoundingBox dimensions="2" crs="urn:ncar:def:
                     <ns2:LowerCorner>-2765.7567138671875 -266.3303222
                     <ns2:UpperCorner>2684.4456787109375 3233.3802490
                  </ns2:BoundingBox>
                  <ns2:BoundingBox dimensions="2" crs="urn:ogc:def:cr
                     <ns2:LowerCorner>229.85759772794773 20.165244646
                     <ns2:UpperCorner>299.1530119233271 50.1202186954
                  </ns2:BoundingBox>
                  <ns6:GridCRS>
                     <ns6:GridBaseCRS>urn:ncar:def:crs:Lambert_Confo
                     <ns6:GridOrigin/>
                     <ns6:GridOffsets/>
                  </ns6:GridCRS>
               </ns6:SpatialDomain>
               <ns6:TemporalDomain>
                  <ns3:timePosition>2010-08-12T21:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T00:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T03:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T06:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T09:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T12:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T15:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T18:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-13T21:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T00:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T03:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T06:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T09:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T12:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T15:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T18:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-14T21:00:00.000Z</ns3:tin
                  <ns3:timePosition>2010-08-15T00:00:00.000Z</ns3:tin
               </ns6:TemporalDomain>
            </ns6:Domain>
            <ns6:Range>
               <ns6:Field>
                  <ns6:Identifier>wind_speed</ns6:Identifier>
                  <ns6:Definition>
                     <ns2:UOM>m/s</ns2:UOM>
                  </ns6:Definition>
                  <ns6:InterpolationMethods>
                     <ns6:InterpolationMethod>nearest</ns6:Interpola
                  </ns6:InterpolationMethods>
               </ns6:Field>
```

response time: 83ms (3629 bytes)

6. Step three: Perform a getCoverage SOAP request using soapUI:
   - Double click on the "SOAP Request 1" associated with the getCoverageOption. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server.
   - Next, we will need to populate the correct getCoverage information in the left-side window pane. The easiest way to accomplish this is through a simple cut and paste operation. Cut and paste the following information in the left-hand pane. There is one field that will need to be adjusted, and that is the beginPosition and endPosition times. From the describeCoverage info, select the starting (beginPosition) and ending (endPosition) times you'd like to receive data for, the click on green arrow:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns="http://www.opengis.net/wcs/1.1"
xmlns:ns1="http://www.opengis.net/ows/1.1"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDGD-
CONUS-Gridded-LAMP-Visibility</ns1:Identifier>
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84"
dimensions="2">
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this level-
->
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-
04T00:00:00.000Z</ns:BeginPosition>
                  <ns:EndPosition>2010-08-
04T00:00:00.000Z</ns:EndPosition>
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>
               <ns1:Identifier>visibility</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```



http://140.90.8.20:10000/wcs

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope" 
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:ND
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84" dimensio
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this lev
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-14T12:00:00.000Z</ns:Beg
                  <ns:EndPosition>2010-08-14T12:00:00.000Z</ns:EndPo
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>
               <ns1:Identifier>wind_speed</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soa
   <soap:Body>
      <ns6:Coverages xmlns:ns2="http://www.opengis.net/o
         <ns6:Coverage>
            <ns2:Identifier>urn:fdc:mdl-nextgen.nws.noaa
            <ns2:AbstractReferenceBase ns4:role="urn:ogc
         </ns6:Coverage>
         <ns6:CoverageData>
            <xop:Include href="cid:c8c4c8f3-d63b-4f6a-84
         </ns6:CoverageData>
      </ns6:Coverages>
   </soap:Body>
</soap:Envelope>
```

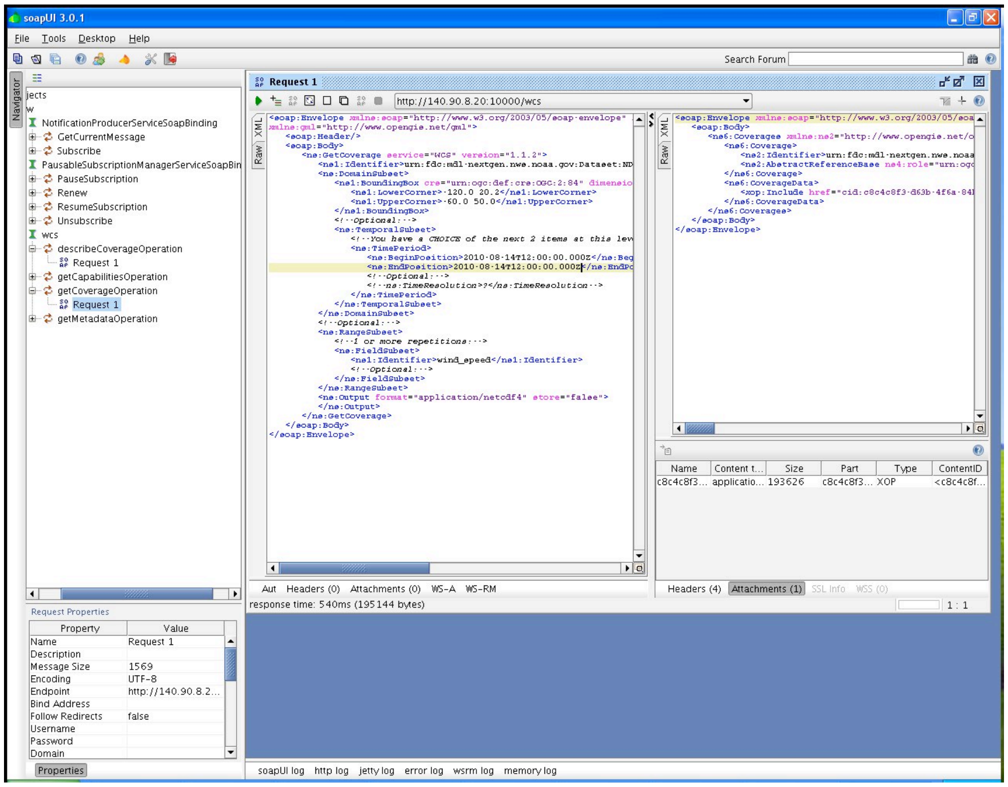
response time: 540ms (195144 bytes)

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

soapUI 3.0.1
File Tools Desktop Help
Search Forum
Navigator
NotificationProducerServiceSoapBinding
GetCurrentMessage
Subscribe
PausableSubscriptionManagerServiceSoapBin
PauseSubscription
Renew
ResumeSubscription
Unsubscribe
wcs
describeCoverageOperation
Request 1
getCapabilitiesOperation
getCoverageOperation
Request 1
getMetadataOperation
Request 1
XML
Raw
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (1) SSL Info WSS (0)
1 : 1
Properties
soapUI log http log jetty log error log wsrm log memory log

7. After execution, a getCoverage soap response will appear in the right-hand pane showing whether or not there was a successful request. If successful, there should be an attachment containing a netCDF4 file. Look for the word “Attachments” towards to lower left-hand corner of the getCoverage soap response (right-side pane).
8. Click once on the word “Attachments”. A panel with “Name”, “Content type”, etc will appear. Below that line is the netCDF4 file.


soapUI 3.0.1
File Tools Desktop Help
Search Forum
Navigator
Request 1
http://140.90.8.20:10000/wcs
XML
Raw
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (1) SSL Info WSS (0)
Request Properties
Properties
soapUI log http log jetty log error log wsrm log memory log

```
jects
w
NotificationProducerServiceSoapBinding
  GetCurrentMessage
  Subscribe
PausableSubscriptionManagerServiceSoapBin
  PauseSubscription
  Renew
  ResumeSubscription
  Unsubscribe
wcs
  describeCoverageOperation
    Request 1
  getCapabilitiesOperation
  getCoverageOperation
    Request 1
  getMetadataOperation
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:ND
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84" dimensio
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this lev
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-14T12:00:00.000Z</ns:Beg
                  <ns:EndPosition>2010-08-14T12:00:00.000Z</ns:EndPo
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>
               <ns1:Identifier>wind_speed</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soa
   <soap:Body>
      <ns6:Coverages xmlns:ns2="http://www.opengis.net/o
         <ns6:Coverage>
            <ns2:Identifier>urn:fdc:mdl-nextgen.nws.noaa
            <ns2:AbstractReferenceBase ns4:role="urn:ogc
         </ns6:Coverage>
         <ns6:CoverageData>
            <xop:Include href="cid:c8c4c8f3-d63b-4f6a-84
         </ns6:CoverageData>
      </ns6:Coverages>
   </soap:Body>
</soap:Envelope>
```

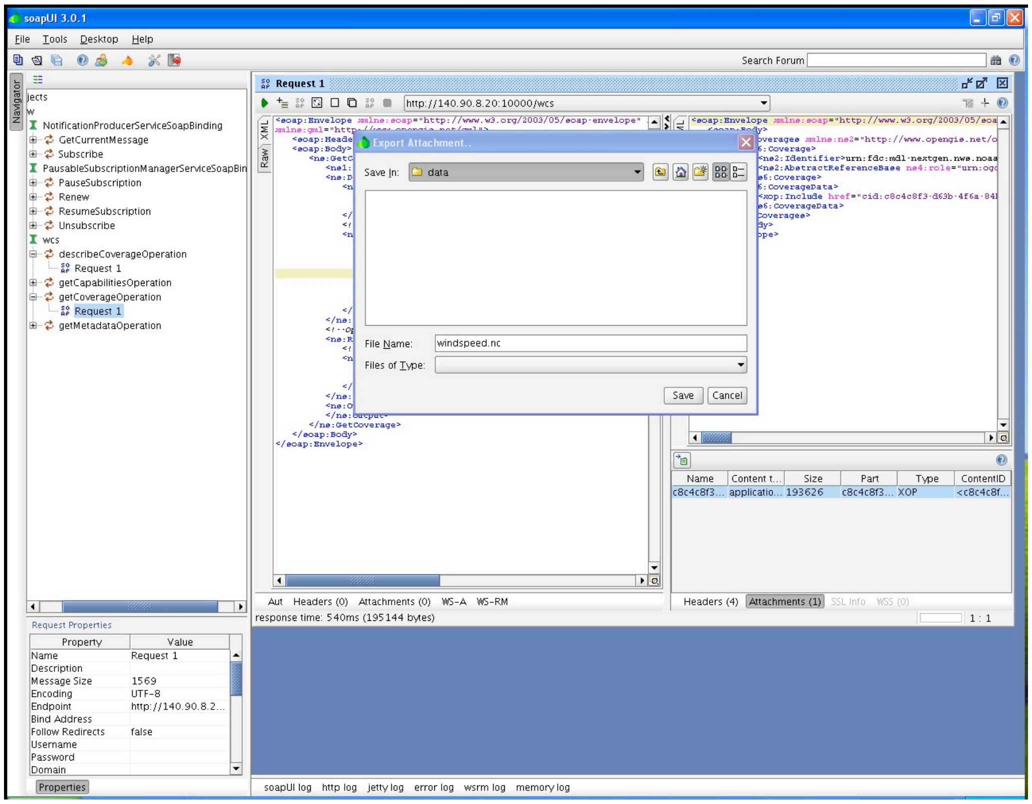
Name	Content t...	Size	Part	Type	ContentID
c8c4c8f3...	applicatio...	193626	c8c4c8f3...	XOP	<c8c4c8f...

response time: 540ms (195144 bytes)

1 : 1

Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

9. Click once anywhere on this line. A green tab will appear that allows for the export of the selected attachment to a file. Click that tab once, and an “Export Attachment” GUI will appear. Save the file to a desired location. A number of programs can be used to analyze the netCDF4 file, including ncview and ncdump.



Export Attachment..

Save In: data

File Name: windspeed.nc

Files of Type:

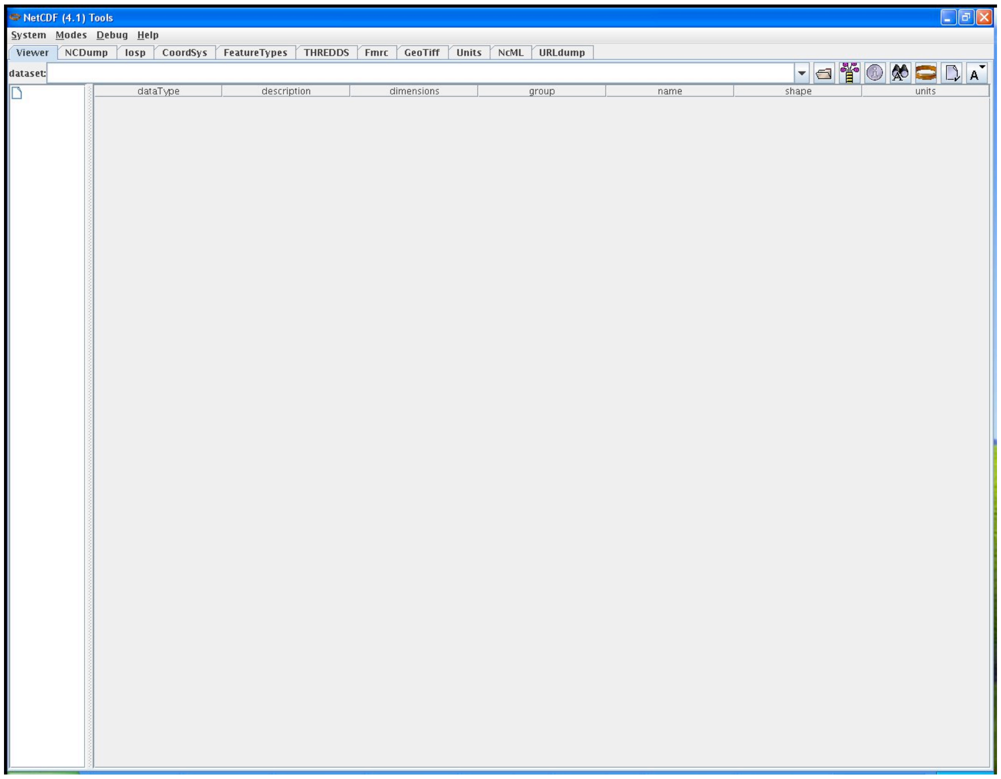
Name	Content t...	Size	Part	Type	ContentID
c8c4c8f3...	applicatio...	193626	c8c4c8f3...	XOP	<c8c4c8f...

response time: 540ms (195144 bytes)

Request Properties

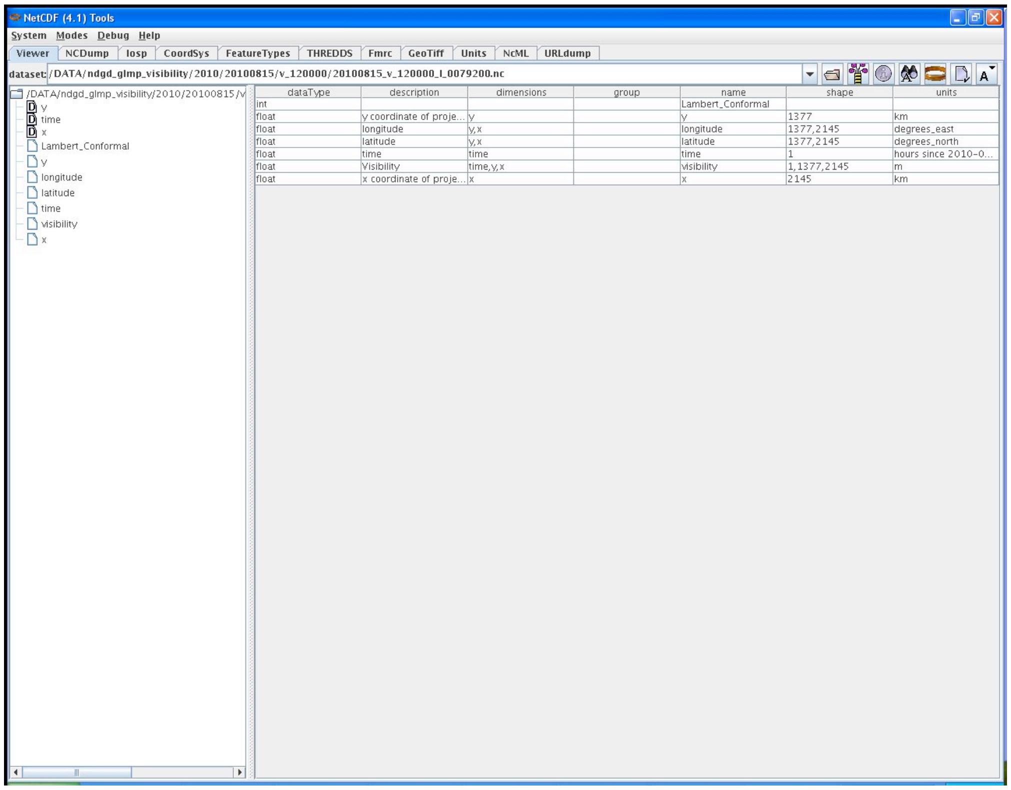
Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

10. A number of programs can be used to view the netCDF4 file, including IDV, toolsUI, ncview and ncdump. We have been using toolsUI version 4.1 (i.e. toolsUI-4.1.jar)
11. Start up toolsUI 4.1. The interface will look something like:


NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump
dataset:

dataType	description	dimensions	group	name	shape	units

12. Click on the open folder icon. Select the image you want to view. Click open, and the toolsUI GUI will look something like:

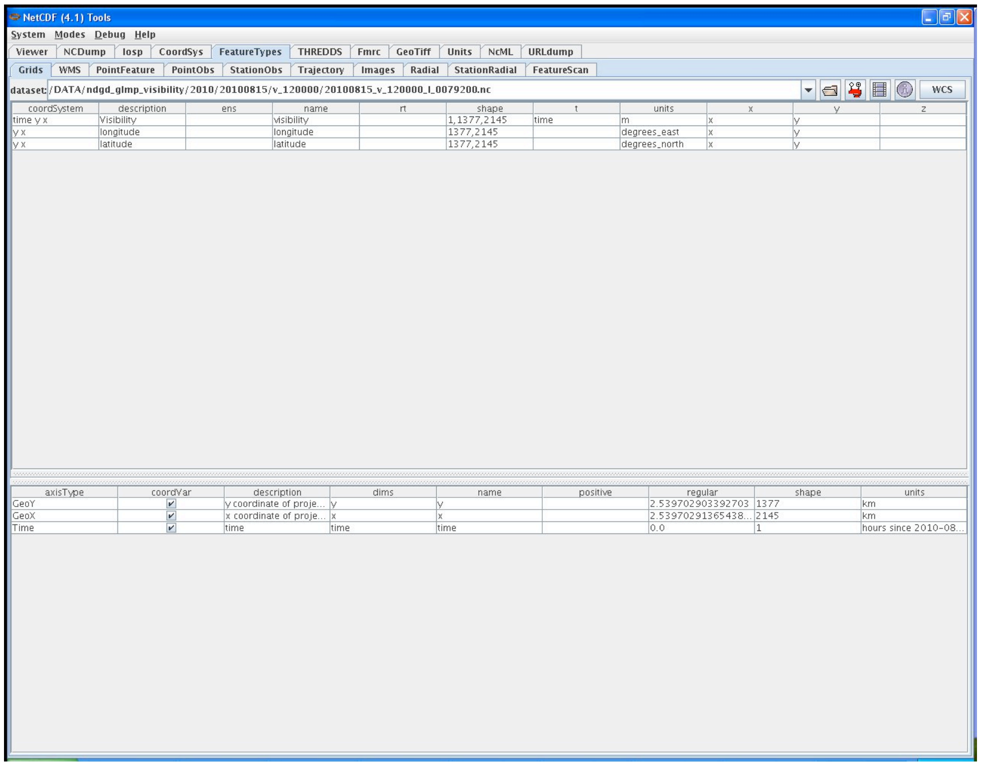

NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump

dataset: /DATA/ndgd_glmp_visibility/2010/20100815/v_120000/20100815_v_120000_l_0079200.nc

/DATA/ndgd_glmp_visibility/2010/20100815/v
- y
- time
- x
- Lambert_Conformal
- y
- longitude
- latitude
- time
- visibility
- x

dataType	description	dimensions	group	name	shape	units
int				Lambert_Conformal		
float	y coordinate of proje...	y		y	1377	km
float	longitude	y,x		longitude	1377,2145	degrees_east
float	latitude	y,x		latitude	1377,2145	degrees_north
float	time	time		time	1	hours since 2010-0...
float	Visibility	time,y,x		visibility	1,1377,2145	m
float	x coordinate of proje...	x		x	2145	km

13. The viewer tab will describe information about the file. In this case, its NDGD Gridded LAMP Visibility.
14. To view the image, click on the FeatureTypes tab, and click on the open folder icon again. Select the image you want to view (20100815_v_120000_0079200.nc in our example) and click open. The toolsUI GUI will look something like:

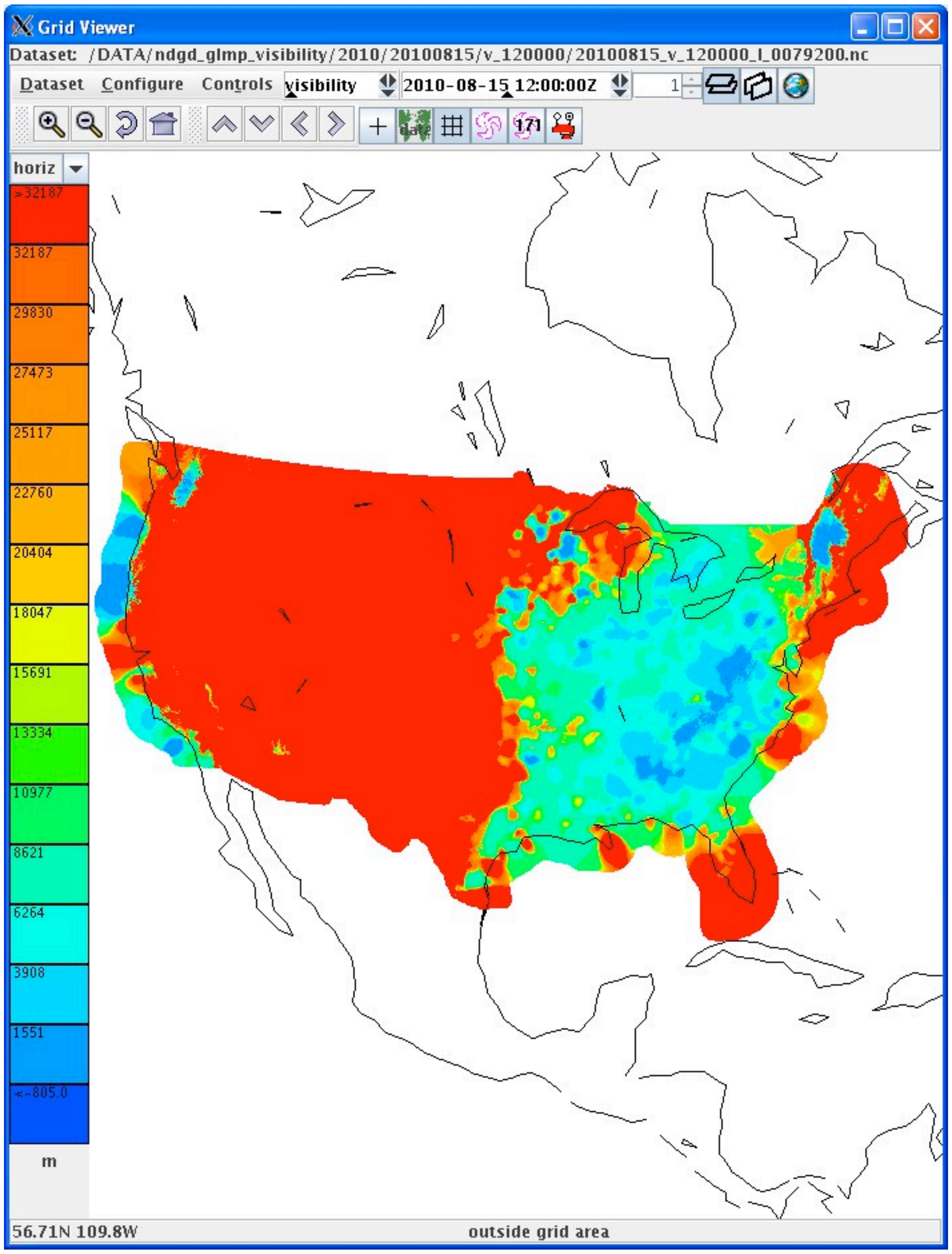

NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump
Grids WMS PointFeature PointObs StationObs Trajectory Images Radial StationRadial FeatureScan
WCS

dataset: /DATA/ndgd_glmp_visibility/2010/20100815/v_120000/20100815_v_120000_l_0079200.nc

coordSystem	description	ens	name	rt	shape	t	units	x	y	z
time y x	Visibility		visibility		1,1377,2145	time	m	x	y	
y x	longitude		longitude		1377,2145		degrees_east	x	y	
y x	latitude		latitude		1377,2145		degrees_north	x	y	

axisType	coordVar	description	dims	name	positive	regular	shape	units
GeoY	✓	y coordinate of proje...	y	y		2.539702903392703	1377	km
GeoX	✓	x coordinate of proje...	x	x		2.53970291365438...	2145	km
Time	✓	time	time	time		0.0	1	hours since 2010-08...

15. Click on the visibility tab and then click the red viewer icon. A new GUI window will appear. Initially, the viewer window should be completely black. Clicking on the red icon one more time will display the image. The viewer GUI should look something like:


Grid Viewer
Dataset: /DATA/ndgd_glmp_visibility/2010/20100815/v_120000/20100815_v_120000_l_0079200.nc
Dataset Configure Controls visibility 2010-08-15 12:00:00Z 1
horiz
>32187
32187
29830
27473
25117
22760
20404
18047
15691
13334
10977
8621
6264
3908
1551
<-805.0
m
56.71N 109.8W
outside grid area

8.4.4.4 *Test Data Reduction and Analysis*

All data collected automatically from the PC during testing will be archived and stored for future reference. All issues that arise during the test will be documented in a Problem Tracking Report (PTR) summarizing the description, criticality and proposed solution of the issue.

8.5 NDGD LAMP Thunderstorm Probability dataset

8.5.1 Introduction

8.5.1.1 Purpose and Scope

The purpose of this test is to demonstrate the way the 4-D Wx Data Cube may operate with National Digital Forecast Database (NDFD) Wind Speed data at initial operating capacity. The publishing and discovery of metadata about the WCS service and the NDFD Wind Speed dataset itself from federated reg/rep agency repositories will be tested. The requirements of the WCS RI are located in Appendix C of the Capability Evaluation Plan. System level requirements in Appendix D. Latencies, although not required, can be measured.

8.5.2 Reference Documents

1. A description of the LAMP elements can be found at http://www.nws.noaa.gov/mdl/gfslamp/docs/elements.shtml
2. A description of the LAMP system can be found at http://www.nws.noaa.gov/mdl/lamp/lamp_info.shtml
3. Access to current LAMP Thunderstorm Probabilities can be found at: http://www.nws.noaa.gov/mdl/gfslamp/tstorm.php

8.5.3 Test Description

8.5.3.1 System Under Test

MDL will be the source system which will be providing NDFD/NDGD data via NOAAnet

8.5.3.2 Test Setup

Prior to the test, the evaluation PC must be loaded with RedHat 5 O/S and configured with OGC compliant software to request and display weather data from the Cube. Unidata's IDV must also be installed to verify that the dataset is available.

a. Test Equipment at the Tech Center
    NAS Simulator, evaluation PC running the IDV, GSD's Testing Portal
b. Test Equipment at remote site
    MDL's WCSRI endpoint with live data

8.5.3.3 Personnel

FAA WJHTC, MDL, GSD

8.5.3.4 Client Location

NWEC lab at WJHTC

8.5.3.5 Server Location

NOAA/NWS/MDL

8.5.4 4.0 Test Conduct

8.5.4.1 Security Concerns

Performance on operational system – to get around this do it at 19Z. Least impact to FTI. Compliant with NIST guided, NOAA/FAA policies on security architecture and relevant information security technologies through a XML gateway.

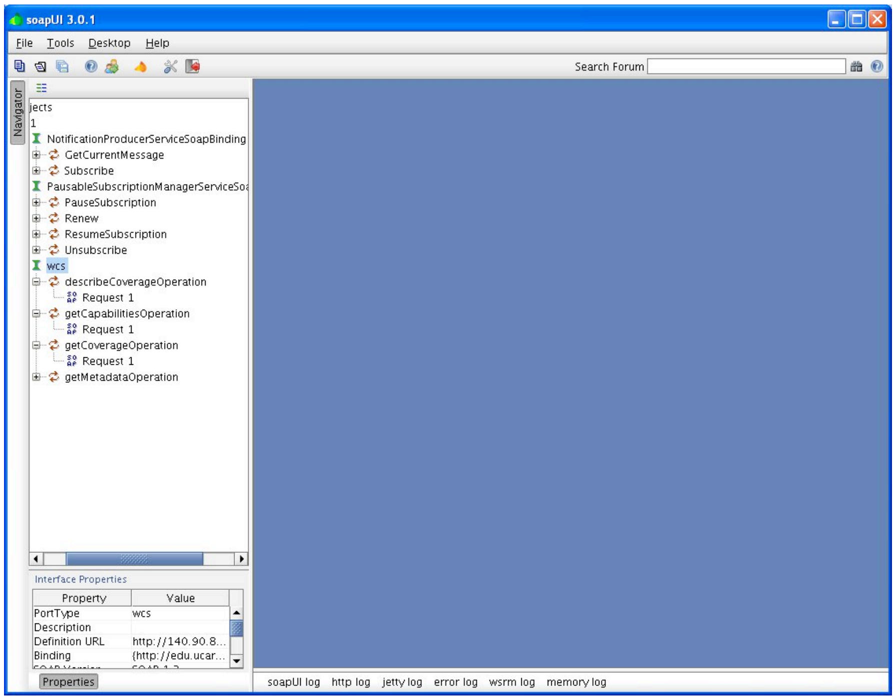
8.5.4.2 Requirements Under Test

- RI shall support all WCSRI 1.1.0/2.0 spec requirements in Appendix C of the current FY10 Capability Evaluation Plan
- Though not required, the system performance (e.g. latency) shall be measured/baselined.

8.5.4.3 Procedures

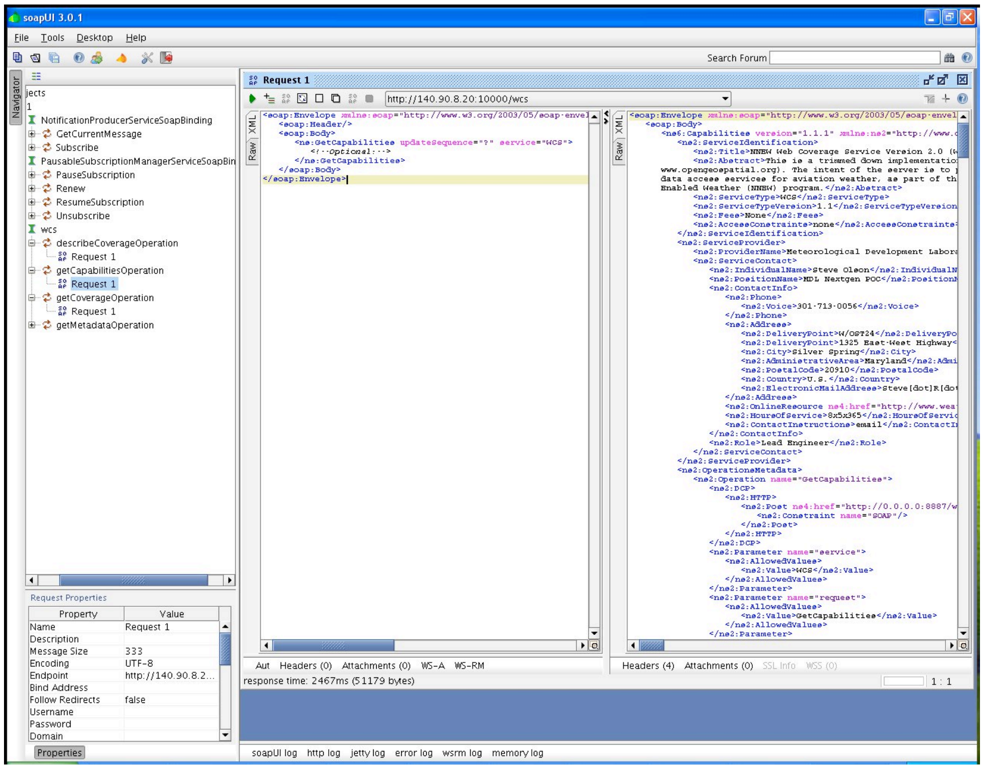
1. Start up soapUI version 3.0+.
   - Import mdl-wcs-soapus-project.xml file if needed. Click on **File**, then **Import Project,** then select file **mdl-wcs-soapui-project.xml , and click on Open**.

2. Click on the "plus" sign and this will expand the **wcs** to include the following SOAP functions:
   - **describeCoverageOperation**
   - **getCapabilitiesOperation**
   - **getCoverageOperation**
   - **getMetadataOperation**



Property	Value
PortType	wcs
Description	
Definition URL	http://140.90.8...
Binding	{http://edu.ucar...

3. Click the "+" symbol for each of these features, and you will reveal a "SOAP Request 1". These soap requests serve as the interface between the user, and our WCS (where the data resides).
4. Step one: Perform a getCapabilities SOAP request using soapUI:
   - Double click on the "SOAP Request 1" associated with getCapabilitiesOperation. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server. There are no fields that need to be populated with this request. This is a simple request to find out what data is available and more background information about our WCS.
   - Click on the green arrow (Submit request to a specified endpoint URL). After execution, a getCapabilities soap response will appear in the right-hand pane indicating some background information about who's offering the service, and what data is available via our WCS.
   - The most important information that is contained in this soap response is the "**ns2:parameter name="Identifier**". This will provide "Values" indicating what data is available. The "urn: value form this soap response is an important input to the second request.
   - Copy the urn information (**urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDGD-CONUS-LAMP-Thunderstorm-Probability**) into the copy and paste buffer.


soapUI 3.0.1

File Tools Desktop Help

Search Forum

```
jects
1
NotificationProducerServiceSoapBinding
  GetCurrentMessage
  Subscribe
PausableSubscriptionManagerServiceSoapBin
  PauseSubscription
  Renew
  ResumeSubscription
  Unsubscribe
wcs
  describeCoverageOperation
    Request 1
  getCapabilitiesOperation
    Request 1
  getCoverageOperation
    Request 1
  getMetadataOperation
```

Request 1

http://140.90.8.20:10000/wcs

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envel
   <soap:Header/>
   <soap:Body>
      <ns:GetCapabilities updateSequence="?" service="WCS">
         <!--Optional:-->
      </ns:GetCapabilities>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envel
   <soap:Body>
      <ns6:Capabilities version="1.1.1" xmlns:ns2="http://www.c
         <ns2:ServiceIdentification>
            <ns2:Title>NNEW Web Coverage Service Version 2.0 (W
            <ns2:Abstract>This is a trimmed down implementatio
      www.opengeospatial.org). The intent of the server is to p
      data access services for aviation weather, as part of th
      Enabled Weather (NNEW) program.</ns2:Abstract>
            <ns2:ServiceType>WCS</ns2:ServiceType>
            <ns2:ServiceTypeVersion>1.1</ns2:ServiceTypeVersion
            <ns2:Fees>None</ns2:Fees>
            <ns2:AccessConstraints>none</ns2:AccessConstraints
         </ns2:ServiceIdentification>
         <ns2:ServiceProvider>
            <ns2:ProviderName>Meteorological Development Labora
            <ns2:ServiceContact>
               <ns2:IndividualName>Steve Olson</ns2:IndividualN
               <ns2:PositionName>MDL Nextgen POC</ns2:PositionN
               <ns2:ContactInfo>
                  <ns2:Phone>
                     <ns2:Voice>301-713-0056</ns2:Voice>
                  </ns2:Phone>
                  <ns2:Address>
                     <ns2:DeliveryPoint>W/OST24</ns2:DeliveryPo
                     <ns2:DeliveryPoint>1325 East-West Highway<
                     <ns2:City>Silver Spring</ns2:City>
                     <ns2:AdministrativeArea>Maryland</ns2:Admi
                     <ns2:PostalCode>20910</ns2:PostalCode>
                     <ns2:Country>U.S.</ns2:Country>
                     <ns2:ElectronicMailAddress>Steve[dot]R[dot
                  </ns2:Address>
                  <ns2:OnlineResource ns4:href="http://www.wea
                  <ns2:HoursOfService>8x5x365</ns2:HoursOfServic
                  <ns2:ContactInstructions>email</ns2:ContactIn
               </ns2:ContactInfo>
               <ns2:Role>Lead Engineer</ns2:Role>
            </ns2:ServiceContact>
         </ns2:ServiceProvider>
         <ns2:OperationsMetadata>
            <ns2:Operation name="GetCapabilities">
               <ns2:DCP>
                  <ns2:HTTP>
                     <ns2:Post ns4:href="http://0.0.0.0:8887/w
                        <ns2:Constraint name="SOAP"/>
                     </ns2:Post>
                  </ns2:HTTP>
               </ns2:DCP>
               <ns2:Parameter name="service">
                  <ns2:AllowedValues>
                     <ns2:Value>WCS</ns2:Value>
                  </ns2:AllowedValues>
               </ns2:Parameter>
               <ns2:Parameter name="request">
                  <ns2:AllowedValues>
                     <ns2:Value>GetCapabilities</ns2:Value>
                  </ns2:AllowedValues>
               </ns2:Parameter>
```

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	333
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

Aut Headers (0) Attachments (0) WS-A WS-RM

Headers (4) Attachments (0) SSL Info WSS (0)

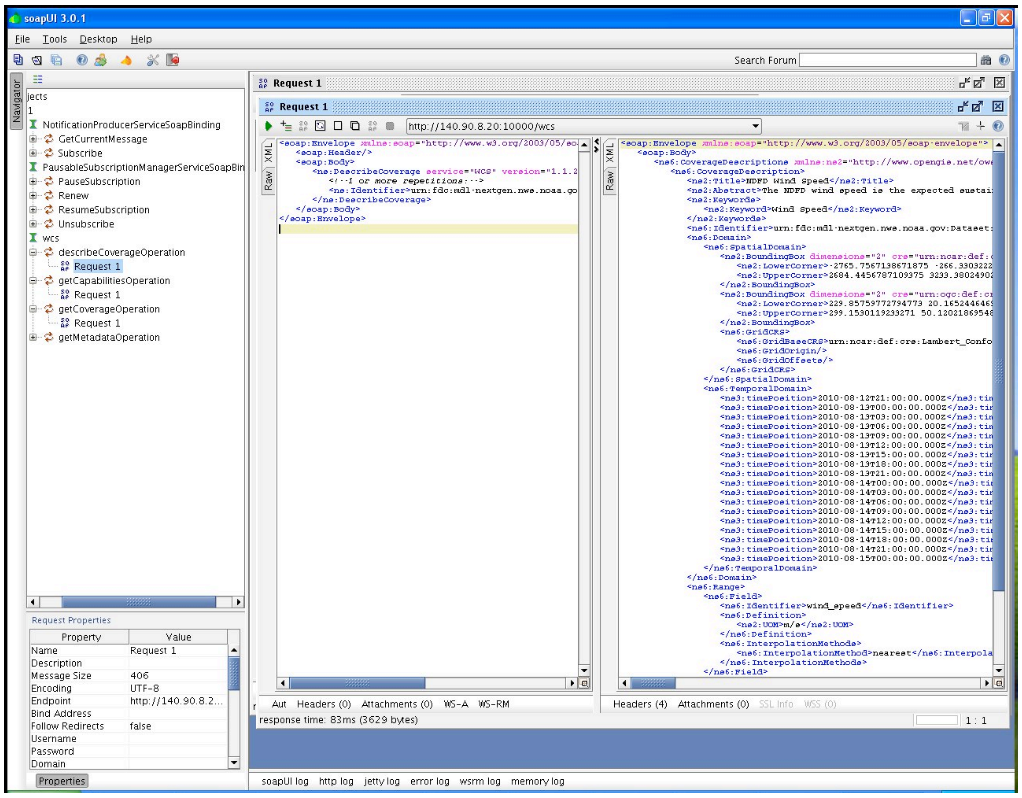
response time: 2467ms (51179 bytes)

1 : 1

5. Step two: Perform a describeCoverage SOAP request using soapUI:
   - Double click on the “SOAP Request 1” associated with the describeCoverageOption. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server.
   - Cut and paste the following information in the left-hand pane, and click on green arrow, or replace the “?” with “urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDGD-CONUS-LAMP-Thunderstorm-Probability”:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-
envelope" xmlns:ns="http://www.opengis.net/wcs/1.1">
   <soap:Header/>
   <soap:Body>
      <ns:DescribeCoverage service="WCS" version="1.1.2">
         <!--1 or more repetitions:-->
      <ns:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDGD-
CONUS-LAMP-Thunderstorm-Probability</ns:Identifier>
      </ns:DescribeCoverage>
   </soap:Body>
</soap:Envelope>
```

1. The screen should look something like this:



http://140.90.8.20:10000/wcs

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/so
   <soap:Header/>
   <soap:Body>
      <ns:DescribeCoverage service="WCS" version="1.1.2
         <!--1 or more repetitions:-->
         <ns:Identifier>urn:fdc:mdl-nextgen.nws.noaa.go
      </ns:DescribeCoverage>
   </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
   <soap:Body>
      <ns6:CoverageDescriptions xmlns:ns2="http://www.opengis.net/ows
         <ns6:CoverageDescription>
            <ns2:Title>NDFD Wind Speed</ns2:Title>
            <ns2:Abstract>The NDFD wind speed is the expected sustai
            <ns2:Keywords>
               <ns2:Keyword>Wind Speed</ns2:Keyword>
            </ns2:Keywords>
            <ns6:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:
            <ns6:Domain>
               <ns6:SpatialDomain>
                  <ns2:BoundingBox dimensions="2" crs="urn:ncar:def:
                     <ns2:LowerCorner>-2765.7567138671875 -266.3303222
                     <ns2:UpperCorner>2684.4456787109375 3233.3802490
                  </ns2:BoundingBox>
                  <ns2:BoundingBox dimensions="2" crs="urn:ogc:def:cr
                     <ns2:LowerCorner>229.85759772794773 20.165244646
                     <ns2:UpperCorner>299.1530119233271 50.12021869548
                  </ns2:BoundingBox>
                  <ns6:GridCRS>
                     <ns6:GridBaseCRS>urn:ncar:def:crs:Lambert_Confo
                     <ns6:GridOrigin/>
                     <ns6:GridOffsets/>
                  </ns6:GridCRS>
               </ns6:SpatialDomain>
               <ns6:TemporalDomain>
                  <ns3:timePosition>2010-08-12T21:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T00:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T03:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T06:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T09:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T12:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T15:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T18:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-13T21:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T00:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T03:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T06:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T09:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T12:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T15:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T18:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-14T21:00:00.000Z</ns3:tim
                  <ns3:timePosition>2010-08-15T00:00:00.000Z</ns3:tim
               </ns6:TemporalDomain>
            </ns6:Domain>
            <ns6:Range>
               <ns6:Field>
                  <ns6:Identifier>wind_speed</ns6:Identifier>
                  <ns6:Definition>
                     <ns2:UOM>m/s</ns2:UOM>
                  </ns6:Definition>
                  <ns6:InterpolationMethods>
                     <ns6:InterpolationMethod>nearest</ns6:Interpola
                  </ns6:InterpolationMethods>
               </ns6:Field>
```

Request Properties

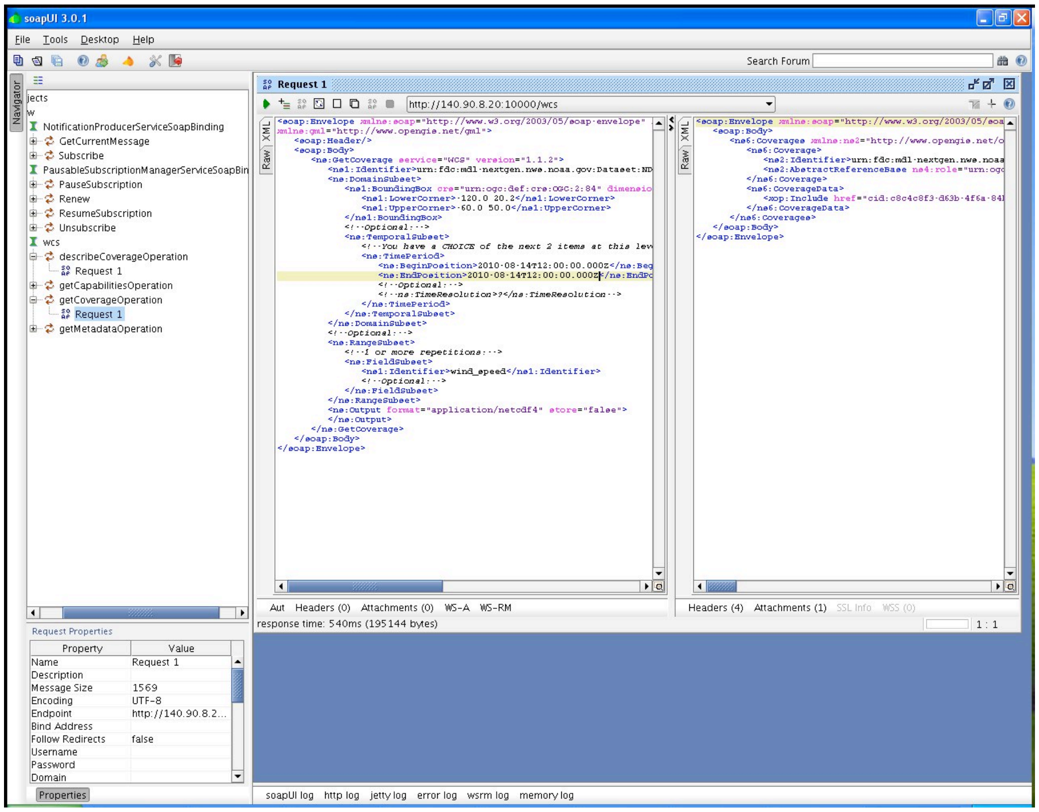
Property	Value
Name	Request 1
Description	
Message Size	406
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

response time: 83ms (3629 bytes)

6. Step three: Perform a getCoverage SOAP request using soapUI:
   - Double click on the "SOAP Request 1" associated with the getCoverageOption. This will open a 2-panel GUI interface. The left-hand side of the interface contains the soap request being sent to our WCS. The right-hand side of the interface contains the soap response from our WCS server.
   - Next, we will need to populate the correct getCoverage information in the left-side window pane. The easiest way to accomplish this is through a simple cut and paste operation. Cut and paste the following information in the left-hand pane. There is one field that will need to be adjusted, and that is the beginPosition and endPosition times. From the describeCoverage info, select the starting (beginPosition) and ending (endPosition) times you'd like to receive data for, the click on green arrow:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:ns="http://www.opengis.net/wcs/1.1"
xmlns:ns1="http://www.opengis.net/ows/1.1"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:NDGD-
CONUS-LAMP-Thunderstorm-Probability</ns1:Identifier>
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84"
dimensions="2">
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this level-
->
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-
04T00:00:00.000Z</ns:BeginPosition>
                  <ns:EndPosition>2010-08-
04T00:00:00.000Z</ns:EndPosition>
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>

<ns1:Identifier>thunderstorm_probability</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```

7. After execution, a getCoverage soap response will appear in the right-hand pane showing whether or not there was a successful request. If successful, there should be an attachment containing a netCDF4 file. Look for the word “Attachments” towards to lower left-hand corner of the getCoverage soap response (right-side pane).


soapUI 3.0.1
File Tools Desktop Help
Search Forum
Request 1
http://140.90.8.20:10000/wcs
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (1) SSL Info WSS (0)
soapUI log http log jetty log error log wsrm log memory log
Properties

Navigator tree:

```
jects
w
NotificationProducerServiceSoapBinding
  GetCurrentMessage
  Subscribe
PausableSubscriptionManagerServiceSoapBin
  PauseSubscription
  Renew
  ResumeSubscription
  Unsubscribe
wcs
  describeCoverageOperation
    Request 1
  getCapabilitiesOperation
  getCoverageOperation
    Request 1
  getMetadataOperation
```

Request:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:gml="http://www.opengis.net/gml">
   <soap:Header/>
   <soap:Body>
      <ns:GetCoverage service="WCS" version="1.1.2">
         <ns1:Identifier>urn:fdc:mdl-nextgen.nws.noaa.gov:Dataset:ND
         <ns:DomainSubset>
            <ns1:BoundingBox crs="urn:ogc:def:crs:OGC:2:84" dimensio
               <ns1:LowerCorner>-120.0 20.2</ns1:LowerCorner>
               <ns1:UpperCorner>-60.0 50.0</ns1:UpperCorner>
            </ns1:BoundingBox>
            <!--Optional:-->
            <ns:TemporalSubset>
               <!--You have a CHOICE of the next 2 items at this lev
               <ns:TimePeriod>
                  <ns:BeginPosition>2010-08-14T12:00:00.000Z</ns:Beg
                  <ns:EndPosition>2010-08-14T12:00:00.000Z</ns:EndPo
                  <!--Optional:-->
                  <!--ns:TimeResolution>?</ns:TimeResolution-->
               </ns:TimePeriod>
            </ns:TemporalSubset>
         </ns:DomainSubset>
         <!--Optional:-->
         <ns:RangeSubset>
            <!--1 or more repetitions:-->
            <ns:FieldSubset>
               <ns1:Identifier>wind_speed</ns1:Identifier>
               <!--Optional:-->
            </ns:FieldSubset>
         </ns:RangeSubset>
         <ns:Output format="application/netcdf4" store="false">
         </ns:Output>
      </ns:GetCoverage>
   </soap:Body>
</soap:Envelope>
```

Response:

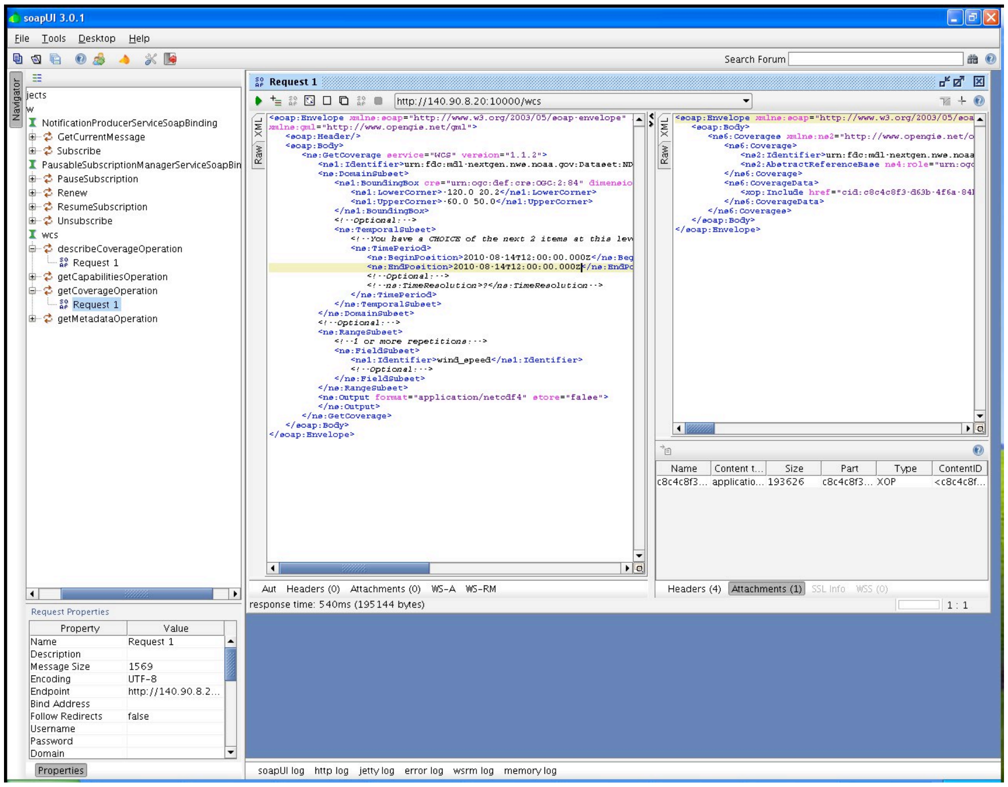
```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soa
   <soap:Body>
      <ns6:Coverages xmlns:ns2="http://www.opengis.net/o
         <ns6:Coverage>
            <ns2:Identifier>urn:fdc:mdl-nextgen.nws.noaa
            <ns2:AbstractReferenceBase ns4:role="urn:ogc
         </ns6:Coverage>
         <ns6:CoverageData>
            <xop:Include href="cid:c8c4c8f3-d63b-4f6a-84
         </ns6:CoverageData>
      </ns6:Coverages>
   </soap:Body>
</soap:Envelope>
```

response time: 540ms (195144 bytes)   1 : 1

Request Properties

Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

8. Click once on the word “Attachments”. A panel with “Name”, “Content type”, etc will appear. Below that line is the netCDF4 file.

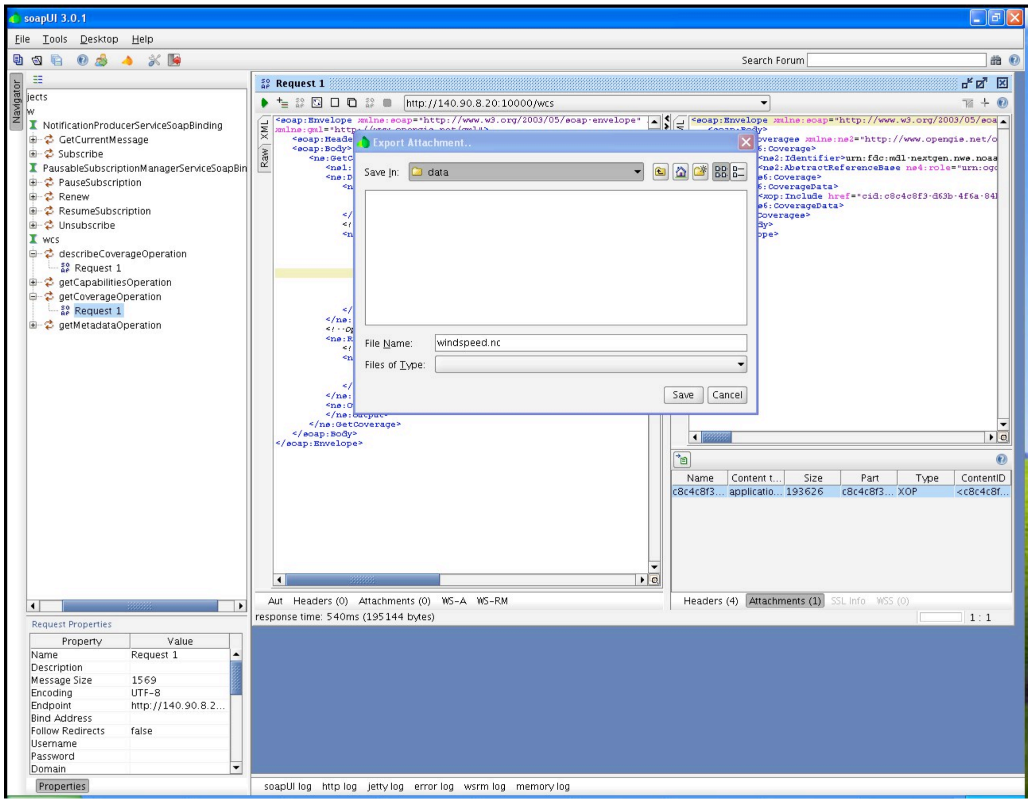

soapUI 3.0.1
File Tools Desktop Help
Search Forum
Request 1
http://140.90.8.20:10000/wcs
Aut Headers (0) Attachments (0) WS-A WS-RM
Headers (4) Attachments (1) SSL Info WSS (0)
soapUI log http log jetty log error log wsrm log memory log
Properties

Request and response XML, navigator tree and Request Properties as in the screenshot above.

Name	Content t...	Size	Part	Type	ContentID
c8c4c8f3...	applicatio...	193626	c8c4c8f3...	XOP	<c8c4c8f...

response time: 540ms (195144 bytes)   1 : 1

9. Click once anywhere on this line. A green tab will appear that allows for the export of the selected attachment to a file. Click that tab once, and an “Export Attachment” GUI will appear. Save the file to a desired location. A number of programs can be used to analyze the netCDF4 file, including ncview and ncdump.



http://140.90.8.20:10000/wcs

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
xmlns:gml="http://www.opengis.net/gml">
    <soap:Header
    <soap:Body>
        <ns:GetC
    ...
        </ns:GetCoverage>
    </soap:Body>
</soap:Envelope>
```

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soa
   ...overages xmlns:ns2="http://www.opengis.net/c
   ...6:Coverage>
   <ns2:Identifier>urn:fdc:mdl-nextgen.nws.noaa
   <ns2:AbstractReferenceBase ns4:role="urn:ogc
   ...6:Coverage>
   ...6:CoverageData>
   <xop:Include href="cid:c8c4c8f3-d63b-4f6a-84
   ...6:CoverageData>
   ...Coverages>
   ...dy>
   ...pe>
```

File Name: windspeed.nc

Files of Type:

Name	Content t...	Size	Part	Type	ContentID
c8c4c8f3...	applicatio...	193626	c8c4c8f3...	XOP	<c8c4c8f...

response time: 540ms (195144 bytes)

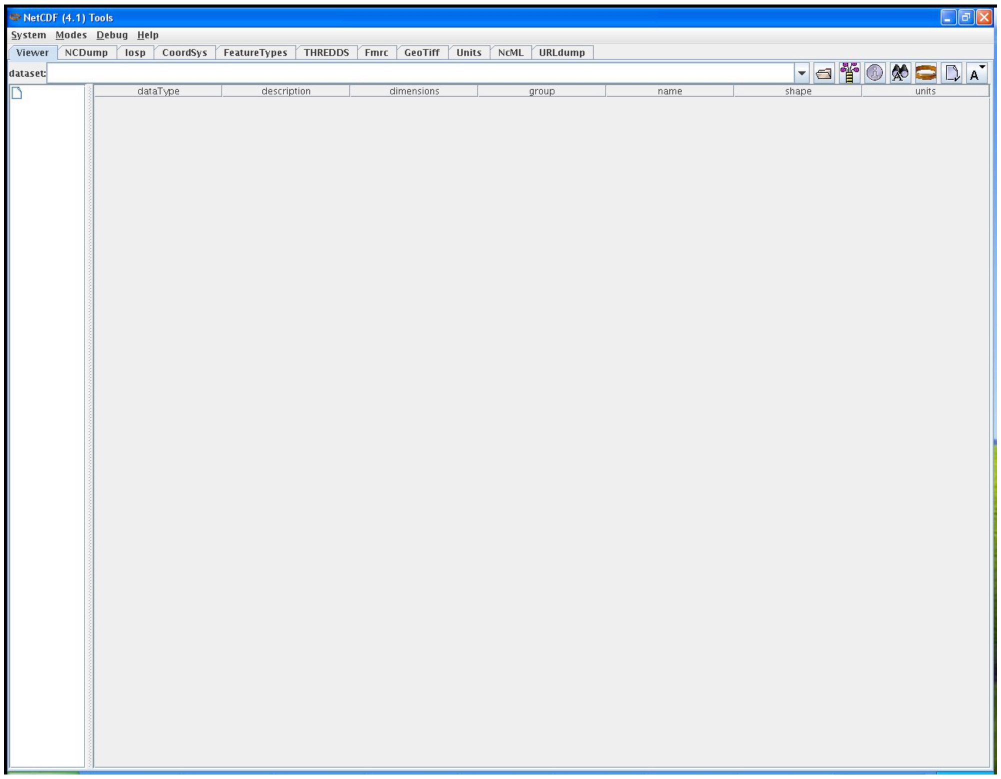
1 : 1

Request Properties

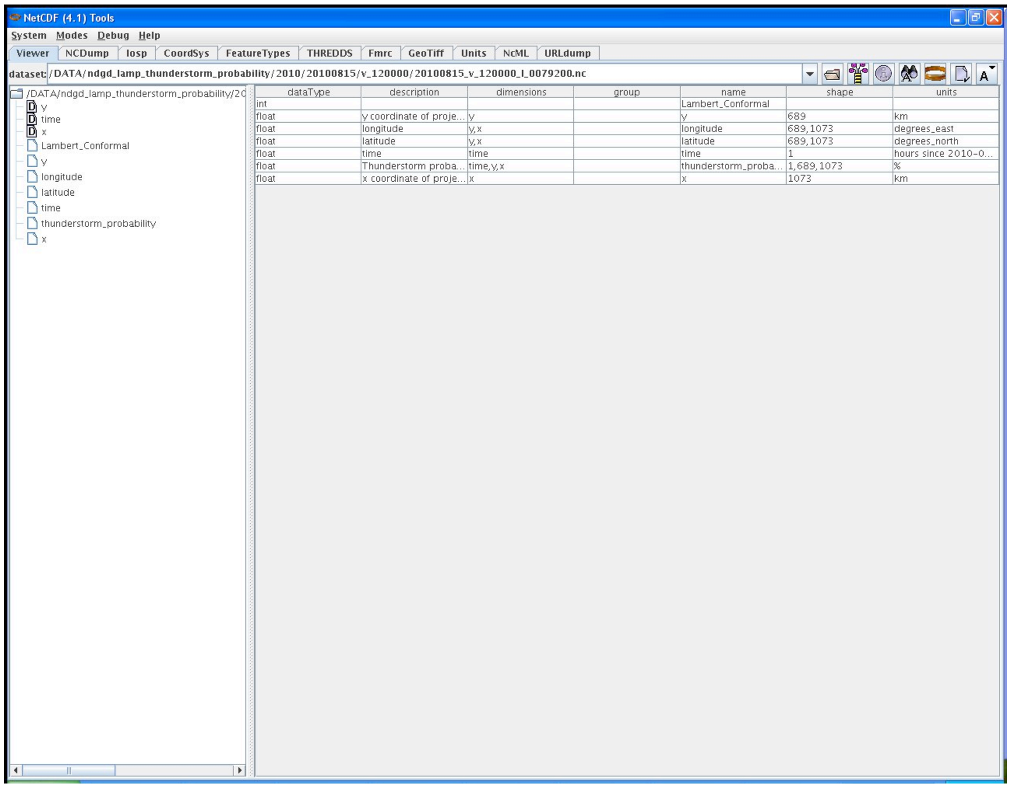
Property	Value
Name	Request 1
Description	
Message Size	1569
Encoding	UTF-8
Endpoint	http://140.90.8.2...
Bind Address	
Follow Redirects	false
Username	
Password	
Domain	

(**NOTE: Wind Speed is shown here**)

10. A number of programs can be used to view the netCDF4 file, including IDV, toolsUI, ncview and ncdump. We have been using toolsUI version 4.1 (i.e. toolsUI-4.1.jar)
11. Start up toolsUI 4.1. The interface will look something like:


NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump
dataset:
dataType description dimensions group name shape units

12. Click on the open folder icon. Select the image you want to view. Click open, and the toolsUI GUI will look something like:

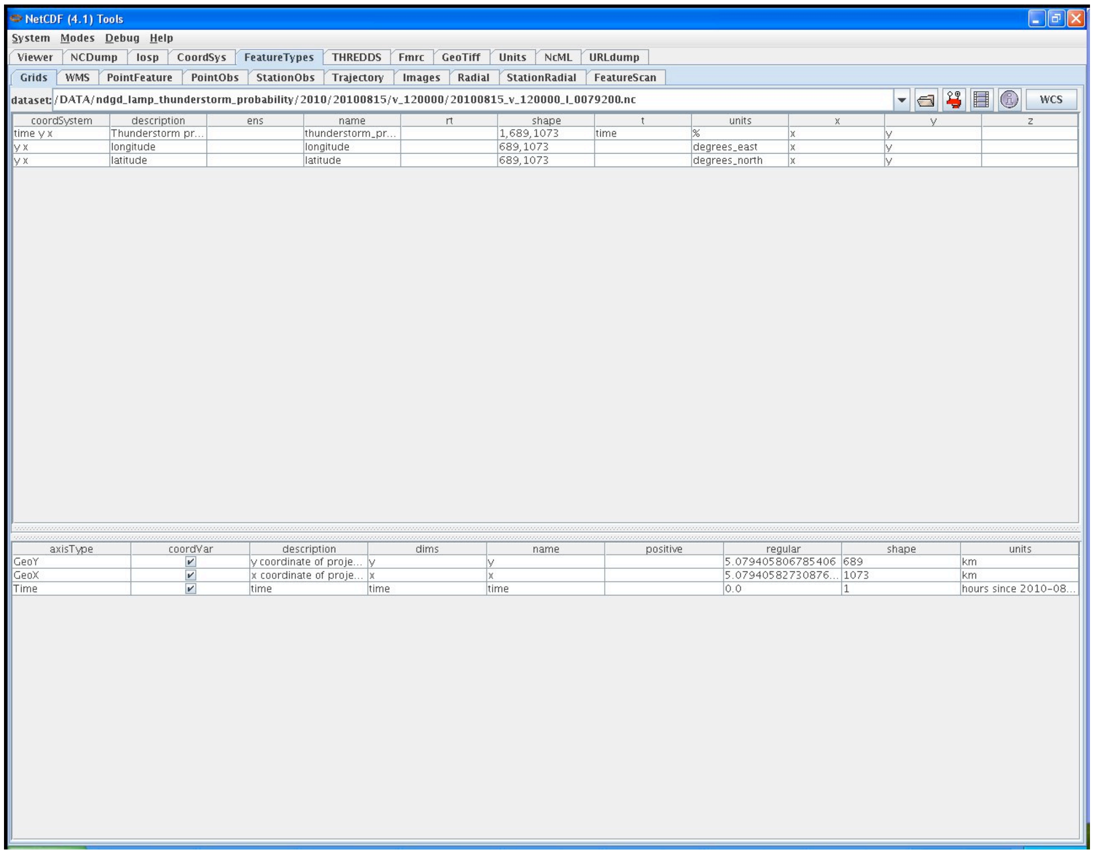

NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump

dataset: /DATA/ndgd_lamp_thunderstorm_probability/2010/20100815/v_120000/20100815_v_120000_l_0079200.nc

```
/DATA/ndgd_lamp_thunderstorm_probability/20
  y
  time
  x
  Lambert_Conformal
  y
  longitude
  latitude
  time
  thunderstorm_probability
  x
```

dataType	description	dimensions	group	name	shape	units
int				Lambert_Conformal		
float	y coordinate of proje...	y		y	689	km
float	longitude	y,x		longitude	689,1073	degrees_east
float	latitude	y,x		latitude	689,1073	degrees_north
float	time	time		time	1	hours since 2010-0...
float	Thunderstorm proba...	time,y,x		thunderstorm_proba...	1,689,1073	%
float	x coordinate of proje...	x		x	1073	km

13. The viewer tab will describe information about the file. In this case, its NDGD LAMP Thunderstorm Probability.
14. To view the image, click on the FeatureTypes tab, and click on the open folder icon again. Select the image you want to view (20100815_v_120000_0079200.nc in our example) and click open. The toolsUI GUI will look something like:

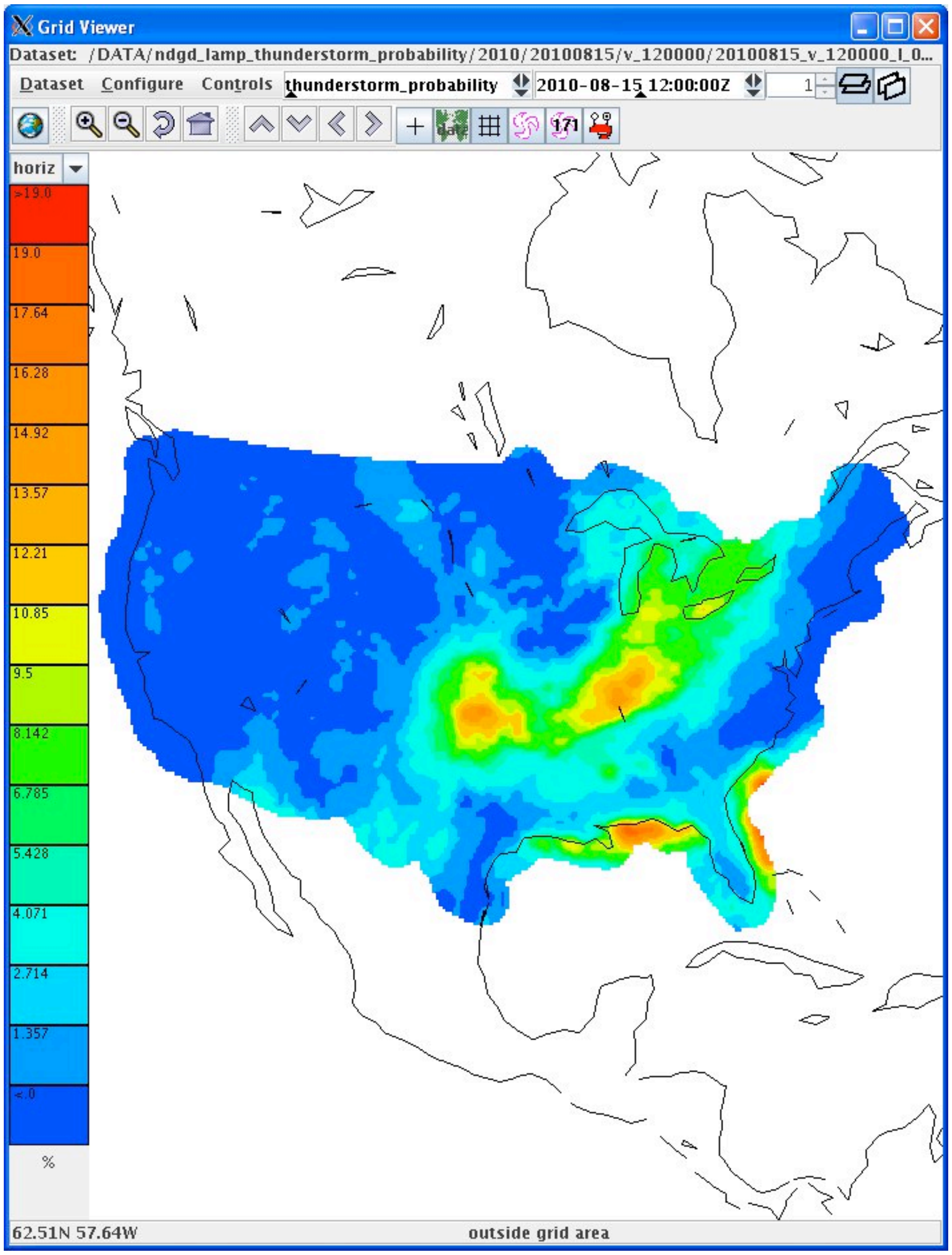

NetCDF (4.1) Tools
System Modes Debug Help
Viewer NCDump Iosp CoordSys FeatureTypes THREDDS Fmrc GeoTiff Units NcML URLdump
Grids WMS PointFeature PointObs StationObs Trajectory Images Radial StationRadial FeatureScan
WCS

dataset: /DATA/ndgd_lamp_thunderstorm_probability/2010/20100815/v_120000/20100815_v_120000_l_0079200.nc

coordSystem	description	ens	name	rt	shape	t	units	x	y	z
time y x	Thunderstorm pr...		thunderstorm_pr...		1,689,1073	time	%	x	y	
y x	longitude		longitude		689,1073		degrees_east	x	y	
y x	latitude		latitude		689,1073		degrees_north	x	y	

axisType	coordVar	description	dims	name	positive	regular	shape	units
GeoY	✓	y coordinate of proje...	y	y		5.079405806785406	689	km
GeoX	✓	x coordinate of proje...	x	x		5.07940582730876...	1073	km
Time	✓	time	time	time		0.0	1	hours since 2010-08...

15. Click on the thunderstorm probability tab and then click the red viewer icon. A new GUI window will appear. Initially, the viewer window should be completely black. Clicking on the red icon one more time will display the image. The viewer GUI should look something like:


Grid Viewer
Dataset: /DATA/ndgd_lamp_thunderstorm_probability/2010/20100815/v_120000/20100815_v_120000_l_0...
Dataset Configure Controls thunderstorm_probability 2010-08-15 12:00:00Z 1
horiz
>19.0
19.0
17.64
16.28
14.92
13.57
12.21
10.85
9.5
8.142
6.785
5.428
4.071
2.714
1.357
< 0
%
62.51N 57.64W
outside grid area

8.5.4.4 *Test Data Reduction and Analysis*

All data collected automatically from the PC during testing will be archived and stored for future reference. All issues that arise during the test will be documented in a Problem Tracking Report (PTR) summarizing the description, criticality and proposed solution of the issue.

8.6 LAMP-based TAFs

8.6.1 Introduction

8.6.1.1 Purpose and Scope

The purpose of this test is to demonstrate the way the 4-D Wx Data Cube may operate with Localized Aviation Model Output Statistics Program (LAMP) -based forecast data at initial operating capacity. The publishing and discovery of metadata about the WFS service and the LAMP data itself from federated reg/rep agency repositories will be tested. The requirements of the WFS RI are located in Appendix C of the Capability Evaluation Plan. System level requirements in Appendix D. Latencies, although not required, can be measured.

8.6.2 Reference Documents

1. A description of the LAMP system can be found at http://www.nws.noaa.gov/mdl/lamp/lamp_info.shtml

8.6.3 Test Description

8.6.3.1 System Under Test

GSD will be the source system which will be providing LAMP-based TAFs via NOAAnet

8.6.3.2 Test Setup

Prior to the test, the evaluation PC must be loaded with RedHat 5 O/S and configured with OGC compliant software to request and display weather data from the Cube. Unidata's IDV must also be installed to verify that the dataset is available.

3.3a Test Equipment at the Tech Center

NAS Simulator, evaluation PC running the IDV, GSD's Testing Portal

3.3b Test Equipment at remote site

GSD's WFSRI endpoint with live data

8.6.3.3 Personnel

FAA WJHTC, MDL, GSD

8.6.3.4 Client Location

NWEC lab at WJHTC

8.6.3.5 Server Location

NOAA/OAR/GSD

8.6.4 Test Conduct

8.6.4.1 Security Concerns

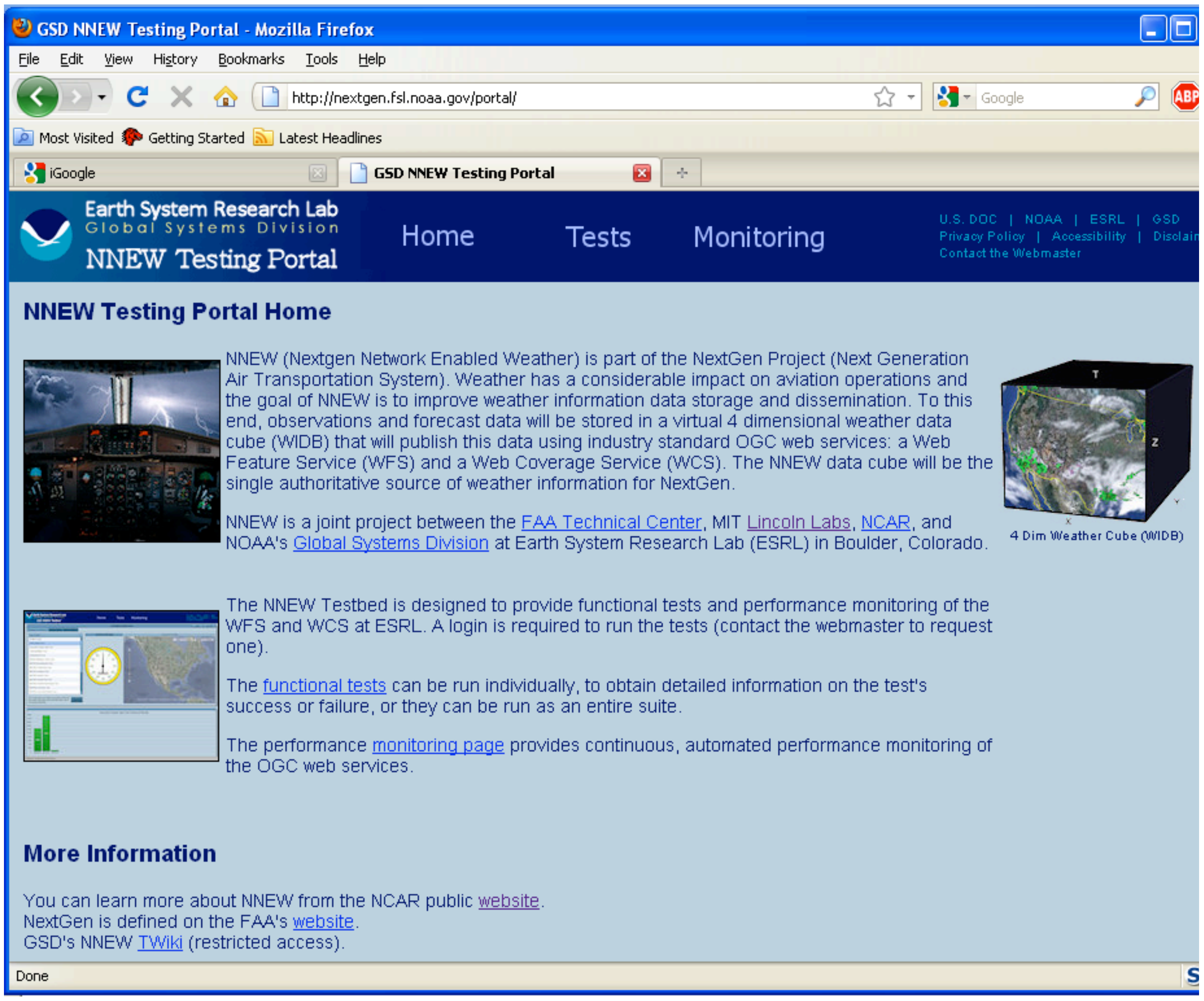
System shall be compliant with NIST guidance, NOAA/FAA policies on security architecture and relevant information security technologies through a XML gateway.

8.6.4.2 Requirements Under Test

- RI shall support all WFSRI 1.1.0/2.0 spec requirements in Appendix C of the current FY10 Capability Evaluation Plan
- Though not required, the system performance (e.g. latency) shall be measured/baselined.

8.6.4.3 Procedures

1. Using a Web browser, such as Internet Explorer or Firefox, access the NNEW WFS Endpoint (http://nextgen.fsl.noaa.gov/portal/):


GSD NNEW Testing Portal - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://nextgen.fsl.noaa.gov/portal/
Most Visited Getting Started Latest Headlines
iGoogle
GSD NNEW Testing Portal
Earth System Research Lab
Global Systems Division
NNEW Testing Portal
Home Tests Monitoring
U.S. DOC | NOAA | ESRL | GSD
Privacy Policy | Accessibility | Disclain
Contact the Webmaster
4 Dim Weather Cube (WIDB)
Done

NNEW Testing Portal Home

NNEW (Nextgen Network Enabled Weather) is part of the NextGen Project (Next Generation Air Transportation System). Weather has a considerable impact on aviation operations and the goal of NNEW is to improve weather information data storage and dissemination. To this end, observations and forecast data will be stored in a virtual 4 dimensional weather data cube (WIDB) that will publish this data using industry standard OGC web services: a Web Feature Service (WFS) and a Web Coverage Service (WCS). The NNEW data cube will be the single authoritative source of weather information for NextGen.

NNEW is a joint project between the FAA Technical Center, MIT Lincoln Labs, NCAR, and NOAA's Global Systems Division at Earth System Research Lab (ESRL) in Boulder, Colorado.

The NNEW Testbed is designed to provide functional tests and performance monitoring of the WFS and WCS at ESRL. A login is required to run the tests (contact the webmaster to request one).

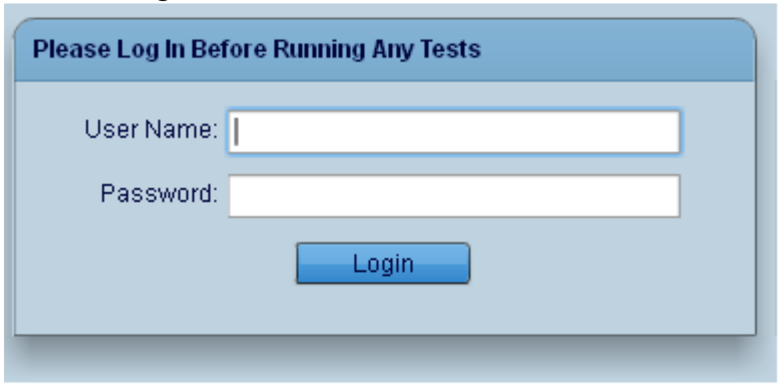
The functional tests can be run individually, to obtain detailed information on the test's success or failure, or they can be run as an entire suite.

The performance monitoring page provides continuous, automated performance monitoring of the OGC web services.

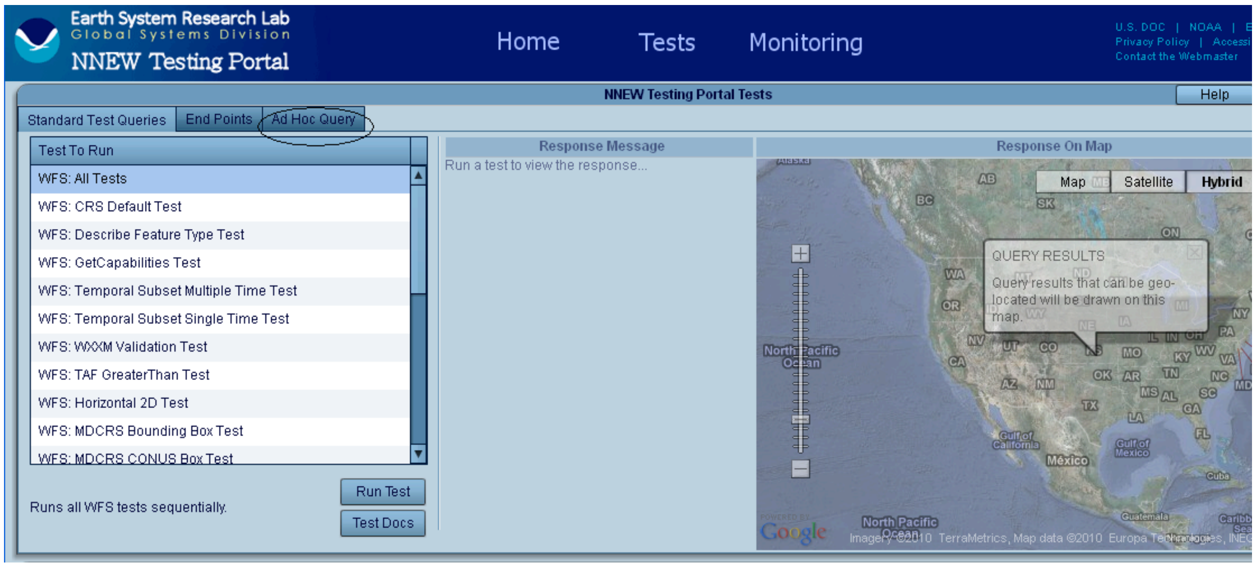
More Information

You can learn more about NNEW from the NCAR public website.
NextGen is defined on the FAA's website.
GSD's NNEW TWiki (restricted access).

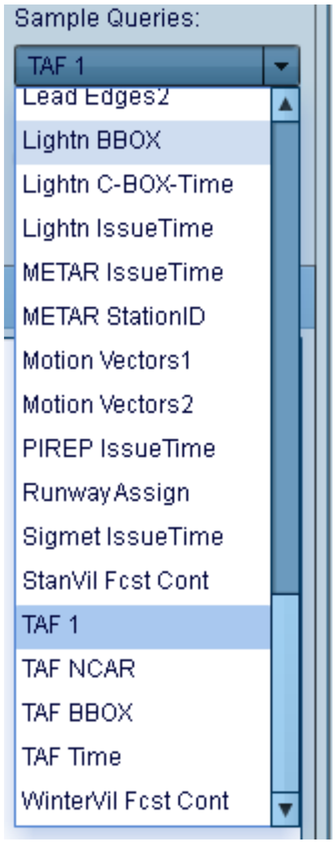
2. In the title bar, there's a link called 'Tests'. Using the mouse, click on that and a security login screen will appear. User name and password will be provided verbally. Press the "Login" button after both fields have been filled in.



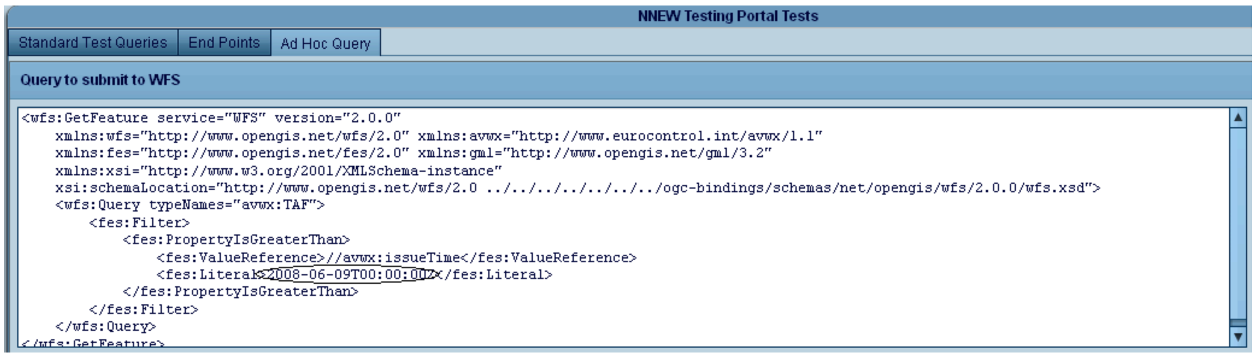
3. If proper name and password have been entered, a new screen will appear in the brower as shown here:


Earth System Research Lab
Global Systems Division
NNEW Testing Portal
Home
Tests
Monitoring
U.S. DOC | NOAA |
Privacy Policy | Accessi
Contact the Webmaster
NNEW Testing Portal Tests
Help
Standard Test Queries
End Points
Ad Hoc Query
Test To Run
WFS: All Tests
WFS: CRS Default Test
WFS: Describe Feature Type Test
WFS: GetCapabilities Test
WFS: Temporal Subset Multiple Time Test
WFS: Temporal Subset Single Time Test
WFS: WXXM Validation Test
WFS: TAF GreaterThan Test
WFS: Horizontal 2D Test
WFS: MDCRS Bounding Box Test
WFS: MDCRS CONUS Box Test
Runs all WFS tests sequentially.
Run Test
Test Docs
Response Message
Run a test to view the response...
Response On Map
Map
Satellite
Hybrid
QUERY RESULTS
Query results that can be geo-located will be drawn on this map.
North Pacific Ocean
México
Gulf of California
Gulf of Mexico
Imagery ©2010 TerraMetrics, Map data ©2010 Europa Technologies, INEGI

4. Using the mouse, select the 'Ad Hoc Query' tab in the upper-left portion of the screen (circled). Again, the screen will change.
5. Using the mouse, under the 'Sample Queries" pull-down menu, select the query labeled 'TAF 1'. The XML document for this query appears in the 'Query to submit to WFS' text-window to the left.



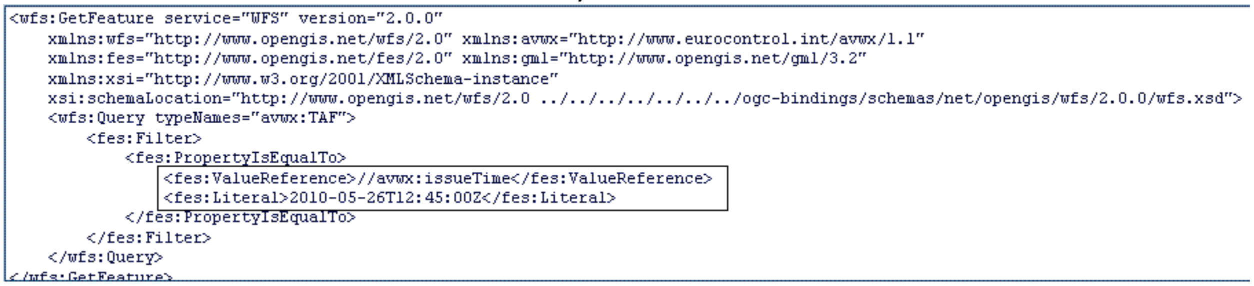
6. Edit the date value found between the <fes:Literal>YYYY-MM-DDTHH:mm:ss</fes:Literal> tags (circled below) to today's date.



```
<wfs:GetFeature service="WFS" version="2.0.0"
    xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:avwx="http://www.eurocontrol.int/avwx/1.1"
    xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/wfs/2.0 ../../../../../../../ogc-bindings/schemas/net/opengis/wfs/2.0.0/wfs.xsd">
    <wfs:Query typeNames="avwx:TAF">
        <fes:Filter>
            <fes:PropertyIsGreaterThan>
                <fes:ValueReference>//avwx:issueTime</fes:ValueReference>
                <fes:Literal>2008-06-09T00:00:00Z</fes:Literal>
            </fes:PropertyIsGreaterThan>
        </fes:Filter>
    </wfs:Query>
</wfs:GetFeature>
```

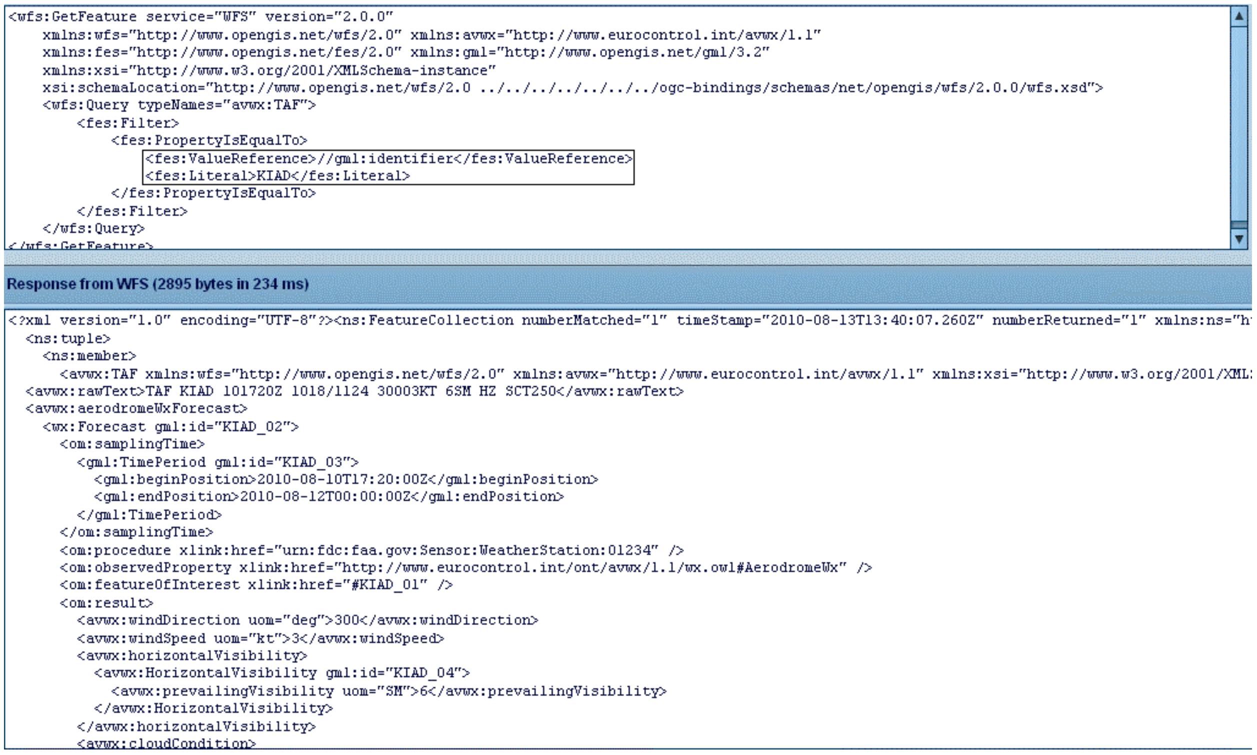
7. With the new date entered, use the mouse to press the 'Run' button on the right-hand side.

This query will retrieve all LAMP-based TAFs issued after the day/time from the WFS endpoint, and the resulting XML document will appear in the "Response from WFS" text window.
8. As in step #5, now select the test labeled "TAF time". The XML query will appear in the text box to the left. The XML Query shown below here too:



```
<wfs:GetFeature service="WFS" version="2.0.0"
    xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:avwx="http://www.eurocontrol.int/avwx/1.1"
    xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/wfs/2.0 ../../../../../../../ogc-bindings/schemas/net/opengis/wfs/2.0.0/wfs.xsd">
    <wfs:Query typeNames="avwx:TAF">
        <fes:Filter>
            <fes:PropertyIsEqualTo>
                <fes:ValueReference>//avwx:issueTime</fes:ValueReference>
                <fes:Literal>2010-05-26T12:45:00Z</fes:Literal>
            </fes:PropertyIsEqualTo>
        </fes:Filter>
    </wfs:Query>
</wfs:GetFeature>
```

9. Change the text enclosed in the box to retrieve any TAF you like. In this example, KIAD (Dulles International) is selected. The ValueReference attribute is changed to "//gml:identifier", and Literal is changed to "KIAD" (or can be any NWS TAF desired). Press "Run" button to send query to WFS. Results of the query are shown in the "Response from WFS" text window, like so:



```
<wfs:GetFeature service="WFS" version="2.0.0"
    xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:avwx="http://www.eurocontrol.int/avwx/1.1"
    xmlns:fes="http://www.opengis.net/fes/2.0" xmlns:gml="http://www.opengis.net/gml/3.2"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/wfs/2.0 ../../../../../../../ogc-bindings/schemas/net/opengis/wfs/2.0.0/wfs.xsd">
    <wfs:Query typeNames="avwx:TAF">
        <fes:Filter>
            <fes:PropertyIsEqualTo>
                <fes:ValueReference>//gml:identifier</fes:ValueReference>
                <fes:Literal>KIAD</fes:Literal>
            </fes:PropertyIsEqualTo>
        </fes:Filter>
    </wfs:Query>
</wfs:GetFeature>
```

Response from WFS (2895 bytes in 234 ms)

```
<?xml version="1.0" encoding="UTF-8"?><ns:FeatureCollection numberMatched="1" timeStamp="2010-08-13T13:40:07.260Z" numberReturned="1" xmlns:ns="h
  <ns:tuple>
    <ns:member>
      <avwx:TAF xmlns:wfs="http://www.opengis.net/wfs/2.0" xmlns:avwx="http://www.eurocontrol.int/avwx/1.1" xmlns:xsi="http://www.w3.org/2001/XML
  <avwx:rawText>TAF KIAD 101720Z 1018/1124 30003KT 6SM HZ SCT250</avwx:rawText>
  <avwx:aerodromeWxForecast>
    <wx:Forecast gml:id="KIAD_02">
      <om:samplingTime>
        <gml:TimePeriod gml:id="KIAD_03">
          <gml:beginPosition>2010-08-10T17:20:00Z</gml:beginPosition>
          <gml:endPosition>2010-08-12T00:00:00Z</gml:endPosition>
        </gml:TimePeriod>
      </om:samplingTime>
      <om:procedure xlink:href="urn:fdc:faa.gov:Sensor:WeatherStation:01234" />
      <om:observedProperty xlink:href="http://www.eurocontrol.int/ont/avwx/1.1/wx.owl#AerodromeWx" />
      <om:featureOfInterest xlink:href="#KIAD_01" />
      <om:result>
        <avwx:windDirection uom="deg">300</avwx:windDirection>
        <avwx:windSpeed uom="kt">3</avwx:windSpeed>
        <avwx:horizontalVisibility>
          <avwx:HorizontalVisibility gml:id="KIAD_04">
            <avwx:prevailingVisibility uom="SM">6</avwx:prevailingVisibility>
          </avwx:HorizontalVisibility>
        </avwx:horizontalVisibility>
        <avwx:cloudCondition>
```

9 WCSRI 2.0 Data Retrieval Tests

9.1 Introduction

Purpose and Scope:

Purpose of these tests is to demonstrate datasets hosted by NCAR for the FY10 Capability Evaluation are available and can be accessed via the test tools. NCAR will be hosting a WCSRI origin server for the gridded data and WFSRI to serve the non-gridded datasets for the FY10 Capability Evaluation. Note, the test procedures documented here will focus on data availability. More detailed procedures showing various ways to request the data are documented in the WCSRI Functional Test Procedures.

NCAR will be provide the following live feeds of gridded products over the public Internet, with appropriate information placed in the Registry/Repository:

Gridded Products Available via WCSRI Service	Available
Rapid Update Cycle (RUC), includes winds, temps, humidity	Yes
Graphical Turbulence Guidance (GTG)	Yes
CIP/FIP (Icing)	Yes
Ceiling	Yes
Visibility	Yes
Flight Category	Yes
NCWF (Convection)	No
GFS (world-wide)	Yes

NCAR will also be providing the following live feeds of non-gridded/WFSRI 2.0 products over the public Internet. The following products will become available over the next few weeks.

Non-Gridded Products Available via WCSRI Service	Available
METARs	August, 2010
PIREPs	August, 2010
TAFs	August, 2010
AIRMETs	August, 2010
SIGMETs(Connective, Non-Connective and Volcanic)	August, 2010
Collaborative Connective Weather Product (CCFP)	August, 2010
G-AIRMETs	August, 2010

9.2 Reference Documents

- Capability Evaluation Plan

9.3 Test Description

9.3.1 System Under Test

- MIT/LL Reg/Rep
- NCAR WCSRI Version 2.0
- MIT/LL WFSRI Version 2.0

9.3.2 Test Setup

The tests will use the data viewer, CubeView, developed by NCAR. It can be installed via Java WebStart from the URL below. CubeView requires Java 1.6 to be on the test system.

http://weather.aero/nnew/fy10/cubeview/
Test Equipment at the Tech Center: A system with Java 1.6 installed
Test Equipment at remote site: NCAR's WCSRI and WFSRI servers
Personnel: FAA WJHTC, NCAR
Client Location: WJHTC
Server Location: NCAR

9.3.3 Test Conduct

9.3.3.1 Security Concerns

9.3.3.2 Requirements Under Test

- The system shall demonstrate any remaining requirements listed in Appendix G of the current FY10 Capability Evaluation plan. Please note, some of these requirements will be addressed in WCSRI T
- The system shall demonstrate data availability for the products.

9.3.3.3 Procedures

1. In a web-browser window, preferably Firefox, enter the following URL: http://weather.aero/nnew/fy10/cubeview/

   Click on the link "**Launch CubeView 2010**"

   This will invoke Java Web Start and download the latest UI (CubeView). It may take sometime to download the application.

   A dialog box asking you for permission to access your system will pop up. Click on "Allow" to move forward.

   Note – You can verify application is from National Center for Atmospheric Research by looking at the signed certificate.

2. CubeView comes pre-configured with data server URLs for the registered products. It takes the user input and forwards a SOAP request to the servers and displays the result in a graphical fashion using a color gradient. CubeView allows user to select time and elevation, where relevant.

3. After CubeView has been launched successfully, the front page will show the map of the continental Unites States, with an overlaid colored temperature grid from recent time at 16,000 ft. To reload temperature values:

   o Select geographic area on the map by dragging the bounding box on the bottom left corner or double-click on the region in the main map window. This should result in reload of the data.
   o The status bar at the bottom on the right side of the map should show "Temperature:Valid". **Errors are shown in red color in the status window.**
   o To verify a different product, e.g., Relative Humidity, Select Relative Humidity from the top-level weather menu.
   o Use the time and altitude selection as appropriate.
4. Select various products as shown in the table below to verify all NCAR datasets are available:

Gridded Product	Menu Option
Rapid Update Cycle (RUC) Temperature Relative Humidy Wind	Weather->Temperature Weather->Relative Humidity Overlays->Wind Barbs
Graphical Turbulence Guidance (GTG)	Weather->Turbulence
CIP/FIP (Icing)	Weather->Icing Weather->Icing Severity Weather->Icing Supercooled Liquid Drops (SLD)
Ceiling	Weather->Ceiling
Visibility	Weather->Visibility
Flight Category	Weather->Flight Category
NCWF (Convection)	
GFS (world-wide)	Weather->Global Temperature Overlays->Global Wind Barbs

9.3.3.4 Test Data Reduction and Analysis

All data collected automatically from the PC during testing will be archived and stored for future reference. All issues that arise during the test will be documented in a Problem Tracking Report (PTR) summarizing the description, criticality and proposed solution of the issue.

10 GSD Data Retrieval

10.1 Introduction

10.1.1 Purpose and Scope

The tests in this section retrieve coverage data from the WCSRI installed at the GSD facility in Boulder. The tests query the WCSRI for information about its capabilities, and use the information in the responses to retrieve a specific coverage. The tests in this section validate the xml responses by extracting information from the xml and using it to retrieve a desired dataset. The tests also show that a data viewing tool developed outside of NNEW can display the returned data, validating the NetCDF or HDF5 format.

10.2 Reference Documents

- FY10 Weather Data Cube Capability Evaluation Plan, Appendix F: Data Flow

10.3 Test Description

The tests in this section are executed using tst-wcs-wx, a testing tool developed by GSD. The tool prompts testers for a URL and coverage, employs the WxConsumer client to handle communications with the WCSRI, and uses Unidata's Integrated Data Viewer (IDV) for data display. TestWcsWx forms requests based on the testers' input, and uses the responses to form the next request:

Tst-wcs-wx displays a list of URLs and asks the testers to select one. The tool sends a getCapabilities request to the URL, and extracts a list of available coverages from the xml response. The testers are prompted to select one of the coverages from the list. A describeCoverage request is sent to the WCSRI, and a list of available times is extracted from the response. The testers then select a time from the list, and they are also asked if they want to specify a bounding box and vertical extent, or use the defaults. A getCoverage request is formed from the testers' input and is sent to the WCSRI. The returned data is stored in a file and the IDV is started, with the file name as input.

10.3.1 System Under Test

The system that will be used for these procedures is the WCSRI installed at GSD. The data sets available from this installation include:

- GFS Air Temperature
- GFS Wind Velocity
- GOES East 11 Micron Infrared (4 Km)
- GOES East 3.9 Micron Infrared (4 Km)
- GOES East 6.7/6.5 Micron Infrared (Water Vapor) (4 Km)
- GOES East Visible (1 Km)

- GOES West 11 Micron Infrared (4 Km)
- GOES West 12 Micron Infrared (4 Km)
- GOES West 3.9 Micron Infrared (4 Km)
- GOES West 6.7/6.5 Micron Infrared (Water Vapor) (8 Km)
- GOES West Visibile (1 Km)
- HRRR 15-minute 2M Temperature
- HRRR 15-minute Echo Top
- HRRR 15-minute Radar Vertically-Integrated Liquid Water
- HRRR 15-minute Surface Pressure
- HRRR 15-minute Vertically-Integrated Liquid Water
- Nexrad Level III Base Reflectivity - 248nm - 0.5deg
- WRF RR Air Temperature
- WRF RR Wind Velocity

10.3.2 Test Setup

System Requirements: ruby, Java 1.6+, Java 3D 1.3.1+, IDV, WxConsumer 3.2, tst-wcs-wx

Setup instructions:

1. Modify /home/nwec/IDV_2.9/runIDV:
   Change -Xmx512m to -Xmx2048

2. Modify /home/nwec/nnew-data/bin/tst-wcs-wx
   a) Set wx_consumer_jar_path
   b) Set idv_run_path
   c) Add desired endpoints to wcsri_endpoint_urls

10.3.3 Personnel

1. Testing personnel at the WJHTC
2. GSD support personnel at the WJHTC and at GSD

10.3.4 Location

The tst-wcs-wx client will be located at the WJHTC; the WCSRI will be located at GSD.

10.4 Test Conduct

10.4.1 Test Procedures and Instructions

This section contains instructions for retrieving data from GSD's WCSRI using tst-wcs-wx. You can use the same procedure to retrieve any of GSD's coverage data by choosing different coverages from the list.

1. cd to /home/nwec/nnew-data/bin.
2. Run tst-wcs-wx.

The program will display a list of endpoints, and you will be asked to select one.

3. Type the number of the GSD production system.

4. The following messages will be displayed:

```
# spawning the getCapabilities request...
$ java -jar /home/nwec/wxConsumer/wx-consumer-wcs-3.4-jar-with-dependencies.jar -e
http://wcs-prod:8280/wcs/soap -o getCapabilities -c none -debug
```

5. The response will include a list of coverages, and you will be asked to select one. For example:

```
0) exit
1) GFS (L-grid) Air Temperature
2) GFS (L-grid) Wind Velocity
3) GOES East 11 Micron Infrared (4 Km)
4) GOES East 3.9 Micron Infrared (4 Km)
5) GOES East 6.7/6.5 Micron Infrared (Water Vapor) (4 Km)
6) GOES East Visible (1 Km)
7) GOES West 11 Micron Infrared (4 Km)
8) GOES West 12 Micron Infrared (4 Km)
9) GOES West 3.9 Micron Infrared (4 Km)
10) GOES West 6.7/6.5 Micron Infrared (Water Vapor) (8 Km)
11) GOES West Visible (1 Km)
12) HRRR 15-minute Radar Vertically-Integrated Liquid Water
13) HRRR 15-minute Vertically-Integrated Liquid Water
14) Nexrad Level III Base Reflectivity - 248nm - 0.5deg (N0Z) for BOX
15) WRF RR Air Temperature
16) WRF RR Wind Velocity

Please select a coverage to retrieve:
```

6. Type the number of the desired coverage.

7. The testing program will send a describeCoverage request to the WCSRI. The response should contain a list of data times which will be displayed, for example:

```
1) 2010-07-05T18:00:00.000Z
2) 2010-07-06T00:00:00.000Z
3) 2010-07-06T06:00:00.000Z
4) 2010-07-06T12:00:00.000Z
```

```
5) 2010-07-06T18:00:00.000Z
6) 2010-07-07T00:00:00.000Z
7) 2010-07-07T06:00:00.000Z
8) 2010-07-07T12:00:00.000Z
9) 2010-07-07T18:00:00.000Z
10) 2010-07-08T00:00:00.000Z
11) 2010-07-08T06:00:00.000Z
12) 2010-07-08T12:00:00.000Z
13) 2010-07-08T18:00:00.000Z
14) 2010-07-09T00:00:00.000Z
15) 2010-07-09T06:00:00.000Z
16) 2010-07-09T12:00:00.000Z

Please select a time to retrieve:
```

8. Type in the number of the desired time.

9. Next, you will be asked:

```
Do you wish to provide a lat/lon bounding box (default: -120,0,-30,90)? [y/n]
```

Type the letter n, to use the default bounding box.

10. Another question will be displayed:

```
Do you wish to provide a vertical extent? (default: 0.12,16.18)? [y/n]
```

Type n, to use the default vertical extent.

11. The test program will then send a getCoverage request to the WCSRI. A status message similar to the following will be displayed:

```
+++ retrieving the coverage+++
# spawning the getCoverage request...
$ java -jar /scratch/wxConsumer/wx-consumer-wcs-3.2-jar-with-dependencies.jar -e
http://wcs-prod:8280/wcs/soap -o getCoverage -c
urn:fdc:fsl.noaa.gov:Dataset:GFS_L_Air_Temperature -f 'Temperature' -t '2010-07-
09T12:00:00.000Z' -b -120,0,-30,90 -v 1,1 -of /tmp/tst_wcs.Temperature.12290.nc
+++
+++  If your IDV hits a Java memory or heap space error or crashes with such an
+++  error, you may have to hack your runIDV script to increase the memory IDV
+++  uses, for example, by turning -Xmx512m to -Xmx1024m.
+++
```

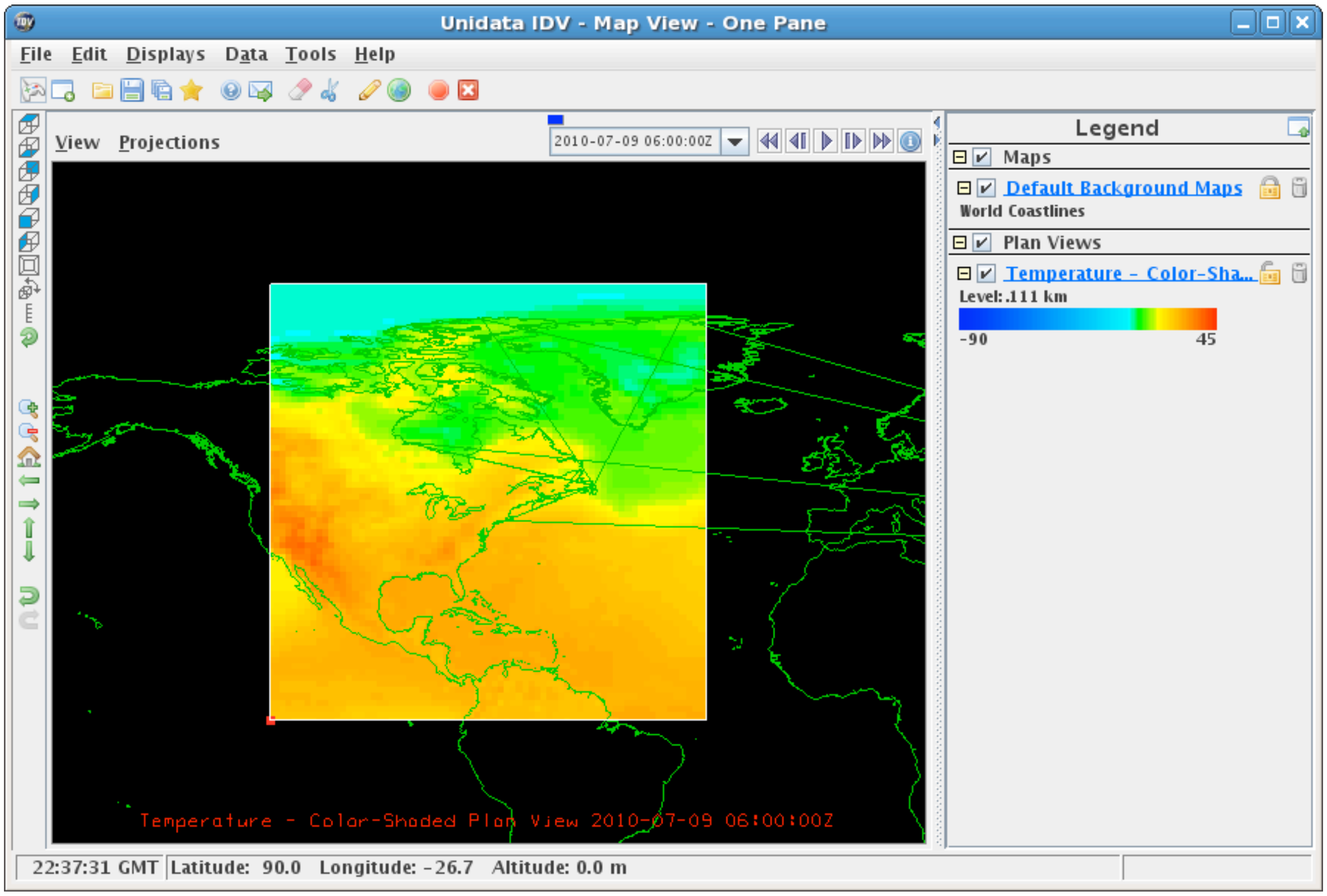
12. If the coverage data is returned successfully, the test program will write it to disk, and automatically start the IDV to display the data. A status message similar to the following will be displayed:

```
# spawning IDV...
$ /scratch/IDV/IDV_2.9/runIDV -data /tmp/tst_wcs.Temperature.12290.nc -display
'Temperature' planviewcolor
```

13. Visually inspect the IDV's display to verify that it contains the desired data. The figure below contains an example of GFS Temperature data displayed in the IDV.

14. To verify the data source (model name, radar location etc) do the following in the IDV display:
    a) Click on 'Data' in the main menu bar
    b) Select the filename, then Edit --> Properties.
    c) Select Metadata. The metadata will contain information identifying the data source.

15. Exit from the IDV


Unidata IDV - Map View - One Pane
File Edit Displays Data Tools Help
View Projections
2010-07-09 06:00:00Z
Legend
Maps
Default Background Maps
World Coastlines
Plan Views
Temperature - Color-Sha...
Level: .111 km
-90
45
Temperature - Color-Shaded Plan View 2010-07-09 06:00:00Z
22:37:31 GMT
Latitude: 90.0 Longitude: -26.7 Altitude: 0.0 m

IDV Display of GFS Temperature Data

11 NEXGEN Weather Processor (NWP)

11.1 Introduction

Since FY07, the FAA's Next Generation Air Transportation System (NextGen) Net-Enabled Weather (NNEW) Program has conducted a yearly IT Demonstration (called Capability Evaluation starting in FY10) to exemplify progress towards the development of the Cube. The FY10 4-D Weather Data Cube Capability Evaluation will be a joint effort of National Oceanic and Atmospheric Administration (NOAA), the FAA and the European Organization for the Safety of Air Navigation (EUROCONTROL) to show progress made towards the development of the 4-D Weather Data Cube.

The primary objective of the FY10 Cube Capability is to simulate operational Cube functionality as closely as possible to show how the Cube will operate at the Initial Operation Capability (IOC) including all applicable Cube standards and available hardware and software infrastructure.

11.1.1 Overview

For the FY10 Data Cube Capability Evaluation, NWP will subscribe to the Cube to access the NEXRAD Level III Product 20 base reflectivity dataset for the (ZNY) New York Air Route Traffic Control Center (ARTCC). NWP will transform the NEXRAD data into the Regional Base Reflectivity Mosaic. The Mosaic will be published to the Cube for subscriptions by users. The NWP Data flow is depicted in Figure 1 below.

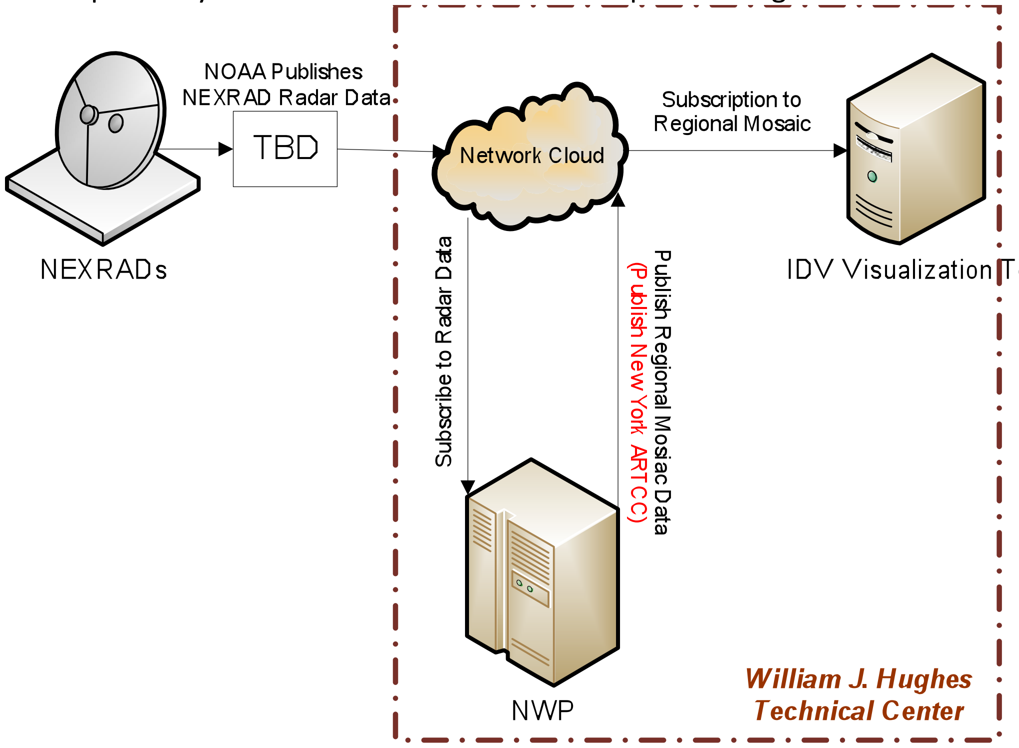


Figure 1 NWP Data Flow

11.1.2 NWP RAMP Functionality Preliminary Design

The initial prototype development will introduce a publish/subscribe SOA architecture as shown in Figure 2. The New York ARTCC regional base reflectivity product (Product 20) will be published into the 4 D cube. These products will be published to a Web Coverage Service (WCS) server by the National Weather Service (NWS). As a back up, William J Hughes Technical Center (WJHTC) will statically populate an instance of the WCS located at the TC with representative radar data.

The WJHTC will build a Web Service client to pull data from the WCS. The request/response service will be used initially since there is no subscription service available at present. This client will query the WCS and use HTTP to get the base reflectivity products formatted as gridded NETCDF 4/CF files. The contents of the gridded NETCDF files will then be converted to radial RPG messages as required by the RAMP input. These RPG messages will be placed into the GFE CM_TCP module receive queue as if they had been received directly from a NEXRAD. The existing RAMP code will be unaware that the data source was the WCS rather than directly from the NEXRADs.

The New York ARTCC base reflectivity mosaic will be built by the RAMP code in an identical manner as it is in operational service today. The completed mosaic will then be published to the cube in NETCDF 4/CF files. The existing DSR Mosaic Dissemination process will also receive the mosaics as required by the DSRs and enable a side by side comparison between the DSR and Web Service pictures. This would leave the existing DSR interface intact but would allow for new users to use the Web Service method.

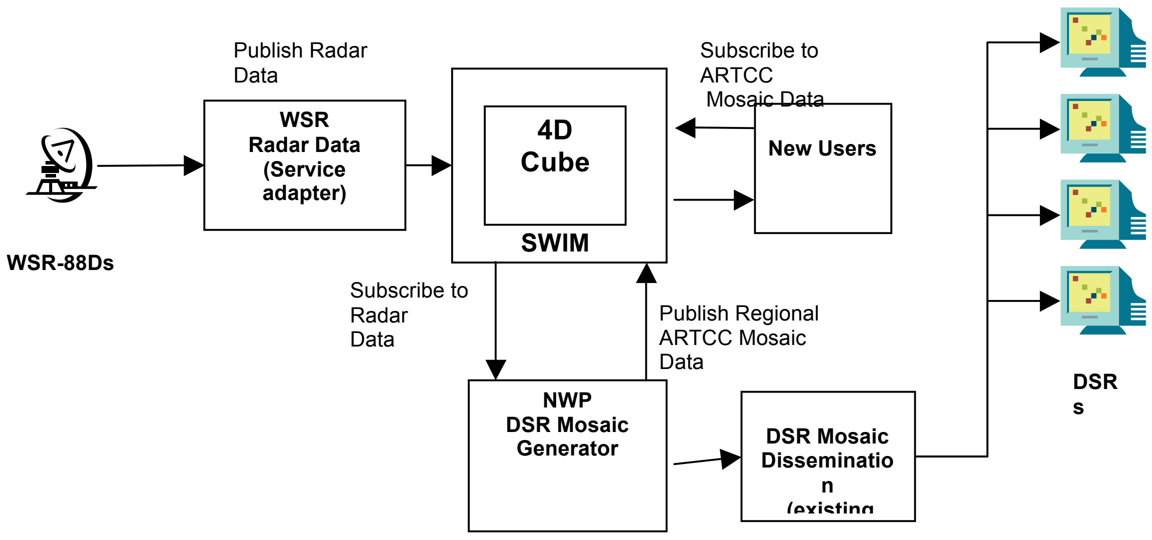

Publish Radar Data
WSR Radar Data (Service adapter)
WSR-88Ds
4D Cube
SWIM
Subscribe to ARTCC Mosaic Data
New Users
Subscribe to Radar Data
Publish Regional ARTCC Mosaic Data
NWP DSR Mosaic Generator
DSR Mosaic Dissemination (existing
DSRs

Figure 2 Data Flow in Next Gen Architecture

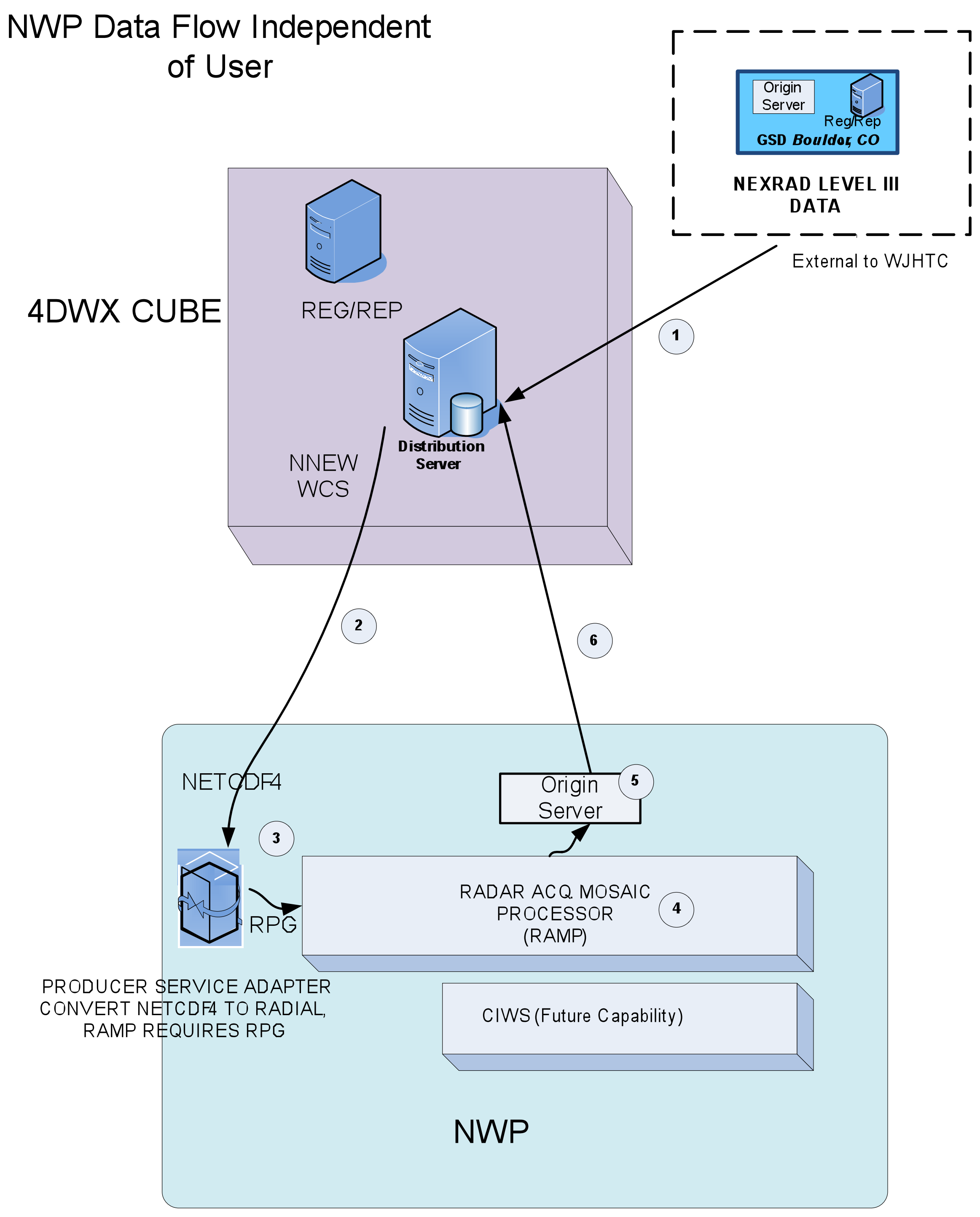
11.2 Reference Document

4-D Weather Data Cube FY10 Capability Evaluation Plan, Version 1.0, April 30, 2010

11.3 Test Description

11.3.1 System Under Test

Figures 3 & 4 below depict the WJHTC data flows:



Note: Numbers are for Process Description only

FIGURE 3 WJHTC NWP DATA FLOW

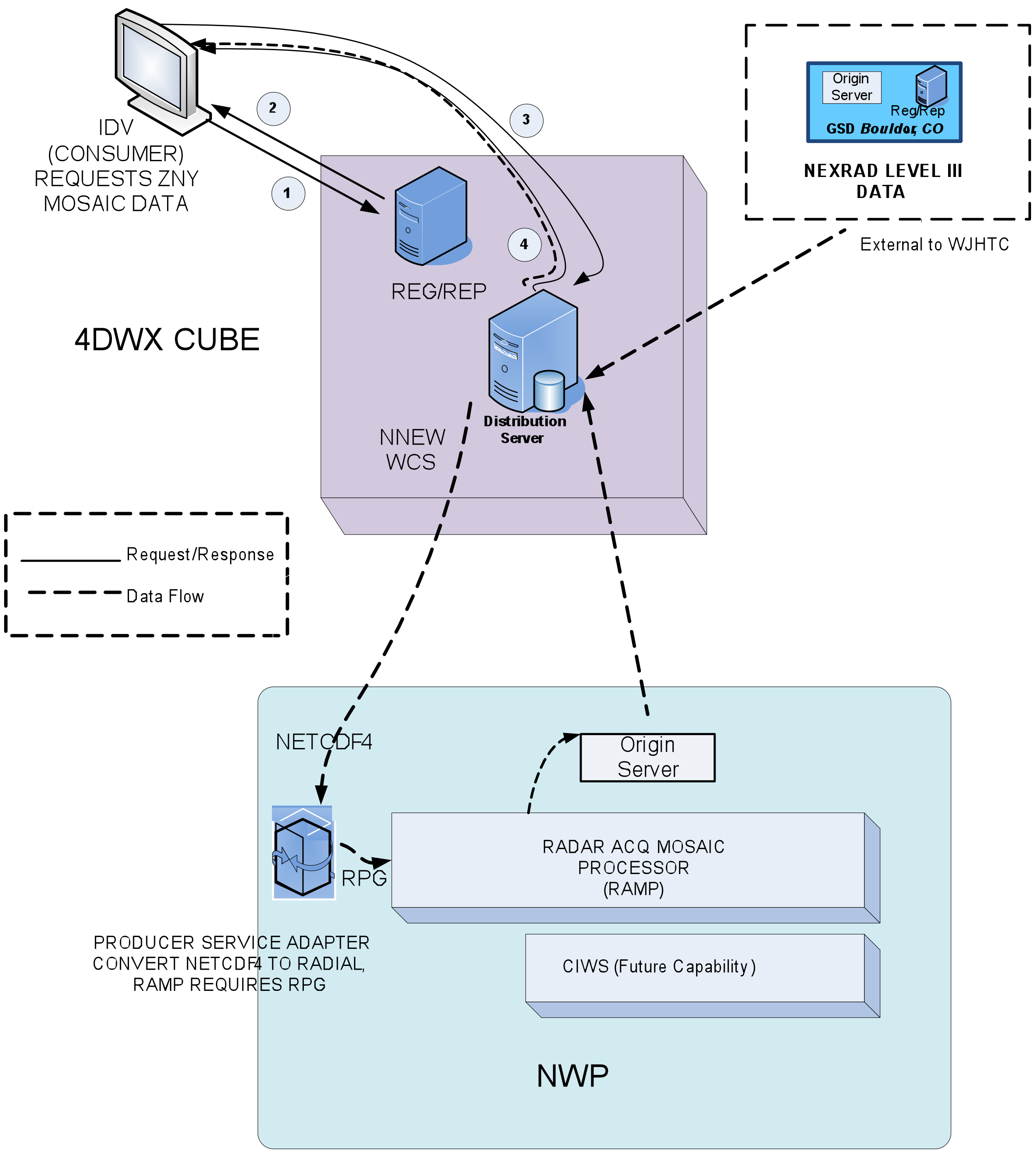


FIGURE 4 WJHTC NWP DATA FLOW
USER REQUESTS ZNY MOSAIC DATA

11.3.1.1 System Data Flow

11.3.1.2 NWP Data Flow Independent of User Request, Figure 2

1. GSD Origin Server sends WSR-88D data in NETCD4 format to 4DWx Cube.
2. WSR radar products are retrieved from the weather cube by the WCSRI client (NETCD gridded format)
3. NWP Service Adaptor Converts NETCDF4 files into JAVA objects and then RPG messages in order for RAMP to build Mosaic
4. RAMP builds WSR-88D Mosaic in RPG in the same manner as at present and outputs data in NETCDF4 format to NWP Origin Server.
5. NWP Origin Server sends Mosaic to 4DWx Cube Distribution Server

Process repeats

11.3.1.3 NWP User Request Process, Figure 3

1. IDV Consumer requests WSR-88D products. Request goes to Reg/Rep in 4DWx Cube
2. Reg/Rep directs IDV to 4Dwx Cube Distribution Server
3. IDV requests WSR-88D products from 4DWx Cube Distribution Server
    Process above has populated the 4DWx Cube Distribution
4. Cube Distribution Server sends MOSAIC data to IDV

Process repeats

11.3.1.4 Data Verification Method-To be completed at later date

The following data sets will be compared using a FLASH VIEWER:

- Original RPG
- Converted RPG to NETCDF4
- NETCDF4 into Cube
- Verify MOSAIC process/Output Pre NetCDF4
- Verify MOSAIC process/Output Post NETCDF4
- NETCDF4 to RPG

Images will be reviewed for intensity, spatial resolution and grid movement. Data will require an active meteorological weather pattern. For meteorological validation a meteorologist will be used to evaluate the input data displays against the MOSAIC output to provide a subjective determination of consistency between the input and output data

11.3.2 Test Equipment

Configured with OGC compliant software to request and display weather data from the Cube, the Evaluation PC is used to demonstrate the implementation of the Web Services. The Evaluation PC running the Red Hat 5 Operating System. The primary application to display weather data will be Integrated Data View (IDV) tool, open source distributed by Unidata. IDV is Javabased software for analyzing and visualizing geosciences data.   RAMP Blade Server IP172.26.162.149 is used for processing.

11.3.3 Test Personnel

Name	*Position*	*Email*	*Phone*
Garth Torok	Electronics Engineer, Project Lead	Garth.Torok@faa.mil	609.485.5184
Robert Owens	Computer Scientist	Bob.CTR.Owns@faa.gov	609.485.5268
Andrew Baron	Computer Scientist	Andrew.CTR.Baron@faa.gov	609.485.5482
Jen Lin	Computer Scientist	Jen.Lin@faa.gov	609.485.8841
Steve Maciejewski	Meteorologist	Steve.Maciejewski@faa.gov	609.485.5950
Melanie Flavin	Electronics Engineer	Melanie.Flavin@faa.gov	609.485.6910

11.3.4 Location

NWP evaluation will be conducted at the FAA WJH Technical Center, Atlantic City, NJ. As part of the evaluation NWP will be accessing the NWS Web Coverage Service (WCS) server.

11.4 Test Conduct

11.4.1 Safety Considerations

Currently none of the systems being utilized for the demonstration are interfacing with live systems.

11.4.2 Requirements under Test

NWP Requirements from 4-D Weather Data Cube FY10 Capability Evaluation Plan, Version 1.0, April 30, 2010, Appendix G: Capability Evaluation Requirements for Supported Datasets/Products:

1. **3.2.2.4.1** (gridded) Shall demonstrate the ingest of NEXRAD Level III weather data from data provider.

2. In addition, WJHTC will be testing the following developed processes:
  -MOSAIC Generation
  -Dissemination of MOSAIC to Origin Server

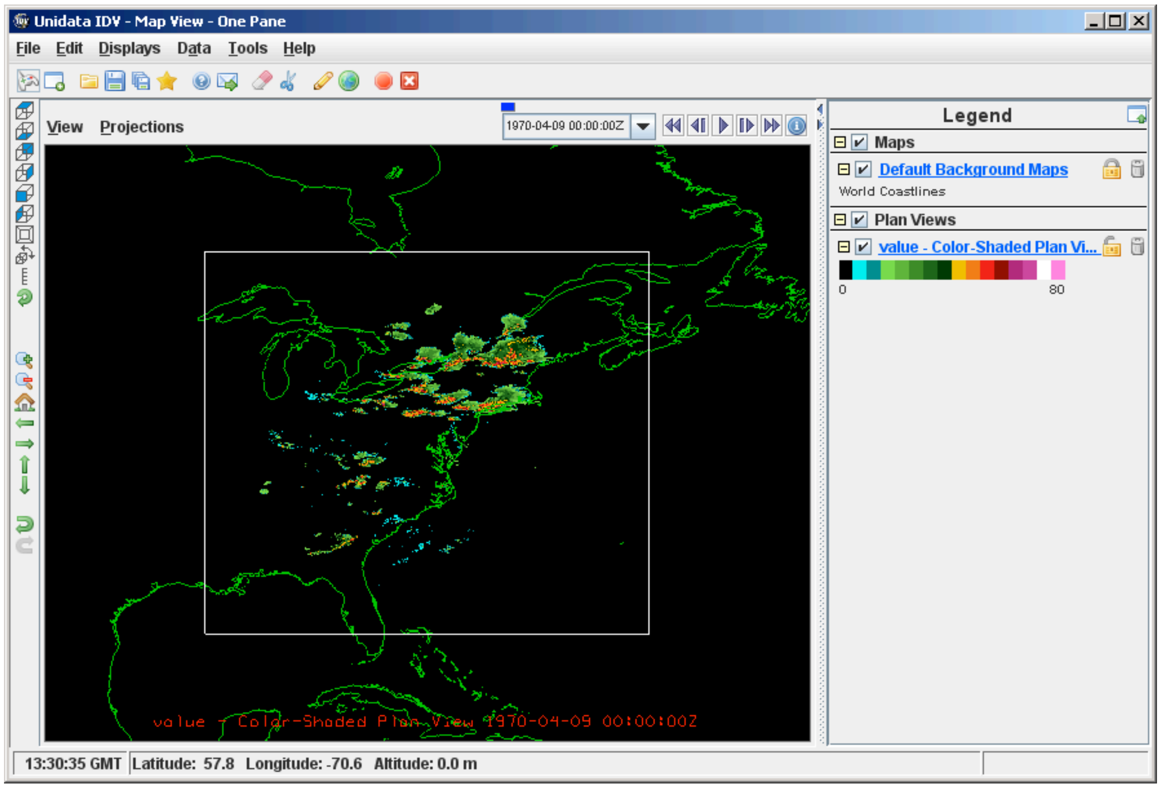
11.4.3 Process Procedures

STEP	***TEST ACTIVITY***	***P/F***	***RESULTS***
10	*RAMP Blade IP192.168.56.18* **ACTION:** Open 4 terminal windows via Putty Username: ramp Password: r@mp!r@mp		
20	Terminal Window 1: **ACTION:** csh		
30	Terminal Window 1: **ACTION:** source .login		
40	Terminal Window 2: ACTION: cd soaRamp		Start up the SOA producer. The producer will move files from the NWPOUT directory to the staging directory of the WCSRI Origin Server.
50	Terminal Window 2: ACTION: ./runsoa producer		
60	Terminal Window 1: **ACTION:** cd /usr/local/share/NWP		
70	Terminal Window 1: **ACTION:** ls –l *		Verify all directories are empty
80	Terminal Window 1: **ACTION: cd $BX**		Move to executable directory
90	Terminal Window 1: **ACTION: ./RAMPUP**		RAMP shared memory regions created RAMP message Queues created RAMP tasks begin execution

STEP	TEST ACTIVITY	P/F	RESULTS
100	Terminal Window 1: **ACTION: ps**		Verify RAMP tasks are executing 1. fiw_wsrmgr 2. z24acfmos 3. z25dsrmos 4. z29mosinpt 5. z31basmos 6. z33hresmos 7. z42unloadmos
110	Terminal Window 1: **ACTION:** **cd /usr/local/share/NWPOUT/WSR**		Verify no output files (NETCDF) exist.
120	Terminal Window 1: **ACTION: ls**		Verify no output files (NETCDF) exist.
130	Terminal Window 3: **ACTION:** cd soaRamp		
140	Terminal Window 3: **ACTION:** ./runsoa mosinput		Start the MOSINPUT application. This converts netcdf files to the RPG message format and places them in the RAMP ingest directory
150	Terminal Window 4: **ACTION:** cd soaRamp		
160	Terminal Window 4: **ACTION:** ./runsoa soaclient		Configuration can be verified by looking in src/soaclient/config/soaclientconfig.xml Up to 2 min delay
170	In the absence of input of Base Reflectivity Files from the origin server: Terminal Window 1: **ACTION:** cd /usr/local/share/NWP/3DEMOS/TC1_7_21_10 ./NWPFEED.csh		Base Reflectivity files will be place in the appropriate directory /usr/local/share/NWP/XXX

STEP	TEST ACTIVITY	P/F	RESULTS
180	Terminal Window 1: **ACTION:** cd /usr/local/share/NWP **ACTION:** ls –l *		Verify Base Reflectivity files begin to appear in the individual Radar directories under /usr/local/share/NWP Note: depending on processing, files may be removed by fiw_wsrmgr task since they are processed upon appearance in the individual radar directories.
190	Terminal Window 1: ACTION: cd $B/log		Move to logs directory
200	Terminal Window 1: ACTION: vi z31basmos.log		Search for “incorporated” This shows that Base reflectivity data has been set into the Base Mosaic Product
210	Terminal Window 1: ACTION: vi z42unloadmos.log		Search for “SUCCESS writing file” This shows that the NETCDF for ZNY ACF has been generated.
220	Terminal Window 1: ACTION: cd /usr/local/share/NWPOUT/WSR		Verify NETCSDF files where written into the directory. Note: If the origin server is present files may only appear briefly before they are removed from the /usr/local/share/NWPOUT directory and placed on the origin server
230	Terminal Window 1: ACTION: ls Repeat ls file may appear and disappear		Verify that files are being removed from RAMP output directory Note: If the origin server is present files may only appear briefly before they are removed from the /usr/local/share/NWPOUT directory and placed on the origin server
240	CUBEVIEW app ****************		Use cubeview to open netcdf files and visualize the generated RAMP mosiac

Below is MOSAIC Image on IDV:


Unidata IDV - Map View - One Pane
File Edit Displays Data Tools Help
View Projections
1970-04-09 00:00:00Z
Legend
Maps
Default Background Maps
World Coastlines
Plan Views
value - Color-Shaded Plan Vi...
0
80
value - Color-Shaded Plan View 1970-04-09 00:00:00Z
13:30:35 GMT Latitude: 57.8 Longitude: -70.6 Altitude: 0.0 m

11.4.4 Test Data Reduction and Analysis

All data collected automatically from the PC during testing will be archived and stored for future reference. All issues that arise during the test will be documented in a Problem Tracking Report (PTR) summarizing the description, criticality and proposed solution of the issues.