ARL-TN-0988 ● JAN 2020

DEVCOM
ARMY RESEARCH
LABORATORY

The Open Geospatial Consortium Web Coverage Service Client Application: A Developer's Guide

by Subing Zeng

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TN-0988 ● JAN 2020

The Open Geospatial Consortium Web Coverage Service Client Application: A Developer's Guide

Subing Zeng
Computational and Information Sciences Directorate, CCDC Army Research Laboratory

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* January 2020	**2. REPORT TYPE** Technical Note	**3. DATES COVERED (From - To)** October 2018–September 2019

4. TITLE AND SUBTITLE

The Open Geospatial Consortium Web Coverage Service Client Application: A Developer's Guide

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Subing Zeng

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

CCDC Army Research Laboratory
ATTN: FCDD-RLC-ED
Adelphi, MD 20783-1138

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TN-0988

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Open Geospatial Consortium Web Coverage Service Client Application (WCSClient) provides the functionality for machine-to-machine retrieval of numerical weather prediction gridded data via the Web Coverage Service, provided by the US Air Force's 557th Weather Wing. The application allows users to request desired data from the available forecast model products by specifying area of interest, forecast reference time, forecast periods, parameters on particular level(s), and the delivery data format.

This developer's guide gives an overview of the Open Geospatial Consortium Web Services, Web Coverage Service, Representational State Transfer transport mechanism, and machine-to-machine data transfer requirements. It then describes software components, application deployment, and execution.

15. SUBJECT TERMS

Open Geospatial Consortium, Web Coverage Service, software component, deployment, execution, client application, machine-to-machine data transfer, forecast model product, gridded data

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Subing Zeng
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**	UU	26	**19b. TELEPHONE NUMBER (Include area code)**
Unclassified	Unclassified	Unclassified			301-394-1961

Contents

List of Figures v

List of Tables v

1. Introduction 1

2. Background 1

2.1 Open Geospatial Consortium Web Services 1

2.2 Web Coverage Service 2

2.3 Representational State Transfer Transport Mechanism 2

2.4 Machine-to-Machine Data Request 3

3. Software Components 3

3.1 Java Package and Class 3

3.2 Property File for Two-Way Authentication 5

3.3 Metadata File for Forecast Model Product 5

3.4 Data Request Definition File 6

3.5 Parameter and Level Definition File 7

4. Multithread and Error Handling 9

5. Application Deployment 9

5.1 Installation 9

5.2 Configuration 9

5.3 Execution 9

6. Conclusion 11

7. References 12

Appendix A. Sample Property File 13

Appendix B. Shortened Version of Metadata for Forecast Model Product 15

List of Symbols, Abbreviations, and Acronyms 18

Distribution List 19

List of Figures

Fig. 1 Java package and relationship of classes 4
Fig. 2 Sample data request definition file 7
Fig. 3 Sample parameter and level definition file 8

List of Tables

Table 1 Java package and main functions.. 5

1. Introduction

The Open Geospatial Consortium (OGC)* Web Coverage Service (WCS)† client application (WCSClient) provides the functionality for machine-to-machine (M2M) retrieval of numerical weather prediction (NWP) gridded data via WCS, provided by the US Air Force's 557th Weather Wing (557WW).‡ The application also allows users to acquire desired data by selecting a forecast model product and defining the area of interest (AOI), forecast reference time/periods, parameters on particular level(s), and the delivery data format.

This report is intended to help the developer understand the technical aspects of the application.

2. Background

This section gives an overview of OGC Web Services, WCS, and the Representational State Transfer (REST) transport mechanism. It also discusses the requirement for establishing M2M connection with the Air Force Weather Agency geographic information system.

2.1 Open Geospatial Consortium Web Services

The OGC Web Services is described per the 557WW Confluence website[1]:

> Open Geospatial Consortium (OGC) Web Services are services defined by the OGC, allowing all kinds of geospatial functionality. They include services for data access, data display and data processing. OGC web service requests are encoded using key-value-pairs (KVP) structures or Extensible Markup Language (XML). The most widely known OGC web services are the Web Feature Service (WFS), Web Coverage Service (WCS), Web Map Service (WMS). At the 557 WW WCS requests are used to retrieve gridded data (Example: GALWEM). WFS requests will be used to retrieve observation data. Finally WMS requests are used to retrieve image or map data.

*The Open Geospatial Consortium (OGC); c2019 [restricted website accessible to US Department of Defense (DOD) personnel with a valid Common Access Card (CAC)]. http://www.opengeospatial.org/.

†The Web Coverage Service (WCS); c2019 [restricted website accessible to DOD personnel with a valid CAC]. https://www.opengeospatial.org/standards/wcs.

‡The 557WW website; c2019 [restricted website accessible to DOD personnel with a valid CAC]. https://weather.af.mil/confluence/display/AFWWEBSTBT/Home

2.2 Web Coverage Service

The WCS is defined per the 557WW Confluence website[2]:

> The OGC Web Coverage Service (WCS) supports electronic retrieval of geospatial data as "coverages" – that is, digital geospatial information representing space/time-varying phenomena. A WCS provides access to coverage data in forms that are useful for client-side rendering, as input into scientific models, and for other clients. The Web Coverage Service provides available data together with their detailed descriptions; defines a rich syntax for requests against these data; and returns data with its original semantics (instead of pictures) which may be interpreted, extrapolated, etc., and not just portrayed. The Web Coverage Service returns coverages representing space/time-varying phenomena that relate a spatio-temporal domain to a (possibly multidimensional) range of properties. As such, WCS focuses on coverages as a specialized class of features and, correspondingly, defines streamlined functionality.

The following WCSs are available at the 557WW:

- GetCapabilities: Delivers an XML-formatted description of the most recent NWP model products available, as well as other pertinent information about the NWP models.
- DescribeCoverage: Delivers XML-formatted descriptions of data coverages, such as location in space and time.
- GetCoverage: Delivers user-defined coverage, either as original data or processed, in some available data format (e.g., GRIB1/GRIB2).
- GetCorridorCoverage: Retrieves weather data or weather conditions along a route in time and space.
- GetPolygonCoverage: Supports polygon or irregular-shaped coverages.

2.3 Representational State Transfer Transport Mechanism

Per Oracle's Java EE 6 Tutorial website[3]:

> **RESTful web services** are built to work best on the Web. Representational State Transfer (REST) is an architectural style that specifies constraints, such as the uniform interface, that if applied to a web service induce desirable properties, such as performance, scalability, and modifiability, that enable services to work best on the Web. In the REST architectural style, data and functionality are considered resources and are

> accessed using **Uniform Resource Identifiers (URIs)**, typically links on the Web. The resources are acted upon by using a set of simple, well-defined operations. The REST architectural style constrains an architecture to a client/server architecture and is designed to use a stateless communication protocol, typically HTTP. In the REST architecture style, clients and servers exchange representations of resources by using a standardized interface and protocol.

The RESTful services at 557WW utilize HTTP's **Get** and **Post** methods.

2.4 Machine-to-Machine Data Request

To request an M2M data transfer via 557WW's OGC Web Services, two-way or mutual Secure Sockets Layer authentication is required between server and client: the user's machine needs to trust the 557WW servers, and the 557WW servers need to trust the user's certificate. Therefore, the user first needs to acquire the DOD-approved Certificate Authority certificates. With these DOD Public Key Infrastructure (PKI) certificates, a software keystore and truststore can be configured properly, then the communication between server and client can be established.

How to obtain the DOD PKI certificates or how to set up the software keystore and truststore is beyond the scope of this report. For detailed information on related topics, visit the 557WW Confluence website.[*]

3. Software Components

The WCSClient consists of a set of Java packages, XML schemas, XML documents, property files, and UNIX/Windows scripts. This section covers each of the software components.

3.1 Java Package and Class

Figure 1 shows the Java package layout and the relationship among classes. Table 1 discusses the main functions of each Java package.

[*] The 557WW information; c2019 Establishing Machine-to-Machine Integration with AFWA GIS Services [restricted website accessible to DOD personnel with a valid CAC]. https://weather.af.mil/confluence/display/SERVICES/Establishing+Machine-to-Machine+Integration+with+AFWA+GIS+Services.

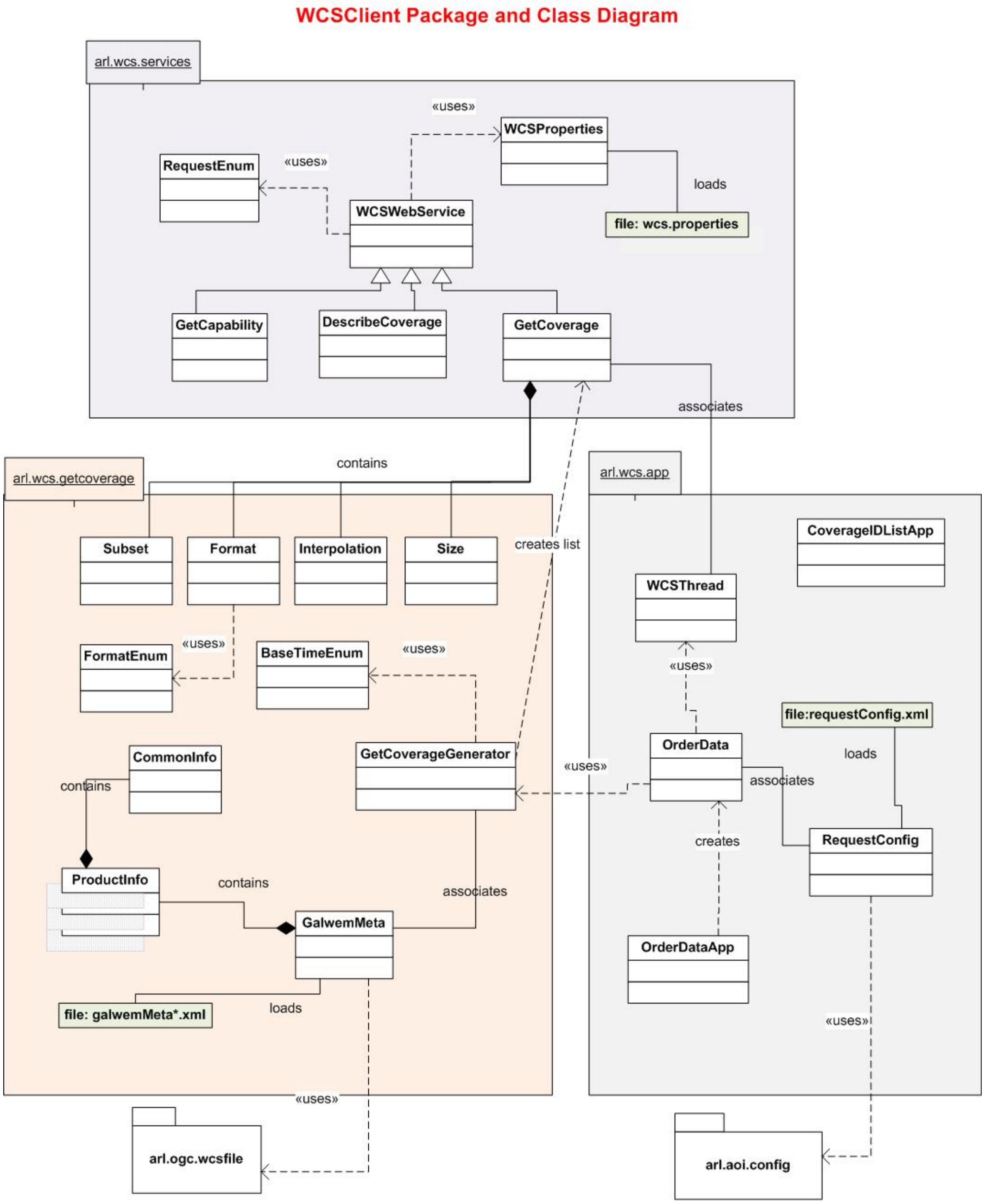


Fig. 1 Java package and relationship of classes

Table 1 Java package and main functions

Package	Description
arl.wcs.service	1. Defines three subclasses related to using 557WW's WCS: GetCapability, DescribeCoverage, and GetCoverage. They are inherited from super class WCSWebService. 2. Defines WCSProperties class, which parses content of a property file. The property values in this file are used by the Java keystore and truststore for M2M data transfer.
arl.wcs.getcoverage	Defines classes for generating data request objects for calling WCS's GetCoverage service.
arl.ogc.wcsfile	A package that is auto-generated by Java Architecture for XML Binding (JAXB) from the GALWEMMeta.xsd schema. This package provides the ability to map Java classes to XML representations and the reverse, and is used for obtaining model product metadata.
arl.aoi.config	A package that is autogenerated by JAXB from the RequestConfig.xsd schema. This package provides the ability to map Java classes to XML representations and the reverse. It is used for obtaining the requested forecast model, AOIs, forecast reference/periods, parameter/level, and delivery format.
arl.wcs.app	1. Contains application for data transfer using multiple threads. 2. Contains application for showing the most recently available coverage IDs for the GALWEM forecast model.

3.2 Property File for Two-Way Authentication

A file with default filename "wcs.properties" is used to store the WCS Uniform Resource Locator (URL) and properties/values, required by the Java keystore and truststore for establishing two-way authentication between the data server and user's machine.

There are two ways to load this property file into the application: 1) include the file "wcs.properties" in the Java's classpath and 2) define a system property "mywida.wcs.configuration" and set the file path as the property value.

For more information about property definition, refer to the sample property file in Appendix A.

3.3 Metadata File for Forecast Model Product

The "GALWEMMeta.xsd" is an XML schema that defines the data structure for storing metadata of the Global Air–Land Weather Exploitation Model

(GALWEM)[*] products, such as data coverage, base reference time, forecast period, parameters, levels, and so on. The data structure is created based on the GALWEM product information returned via the DescribeCoverage service. It is a stable file but may be updated as needed.

Appendix B shows a shortened version of "galwemMeta_0P25DEG.xml", which represents a GALWEM data product with a 0.25° resolution. The information stored in this type of metadata file is used by the GetCoverageGenerator Java class to validate a user's data request.

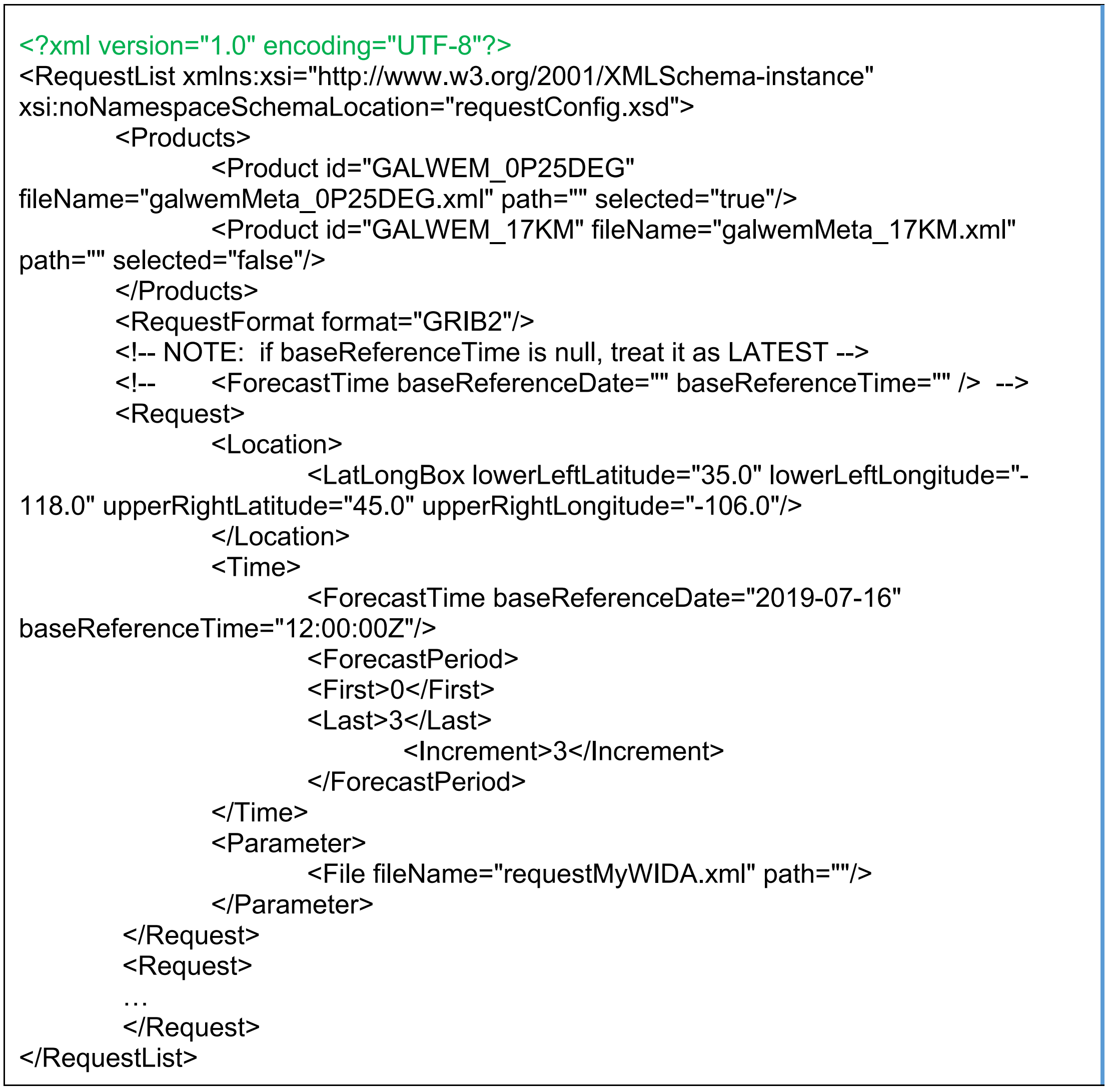
3.4 Data Request Definition File

The "RequestConfig.xsd" is an XML schema that defines the data structure for storing information from the user's data request, such as which GALWEM product is selected, AOIs, base reference time, forecast periods, the delivery data format, and the XML file that specifies desired parameters on particular level(s).

One "Request" element in the XML file contains exactly one "Location", "Time", and "Parameter" element. One or more requests can be defined within this file.

Figure 2 shows a shortened data-request definition file.

[*] The 557WW GALWEM support; c2019 [restricted website accessible to DOD personnel with a valid CAC]. https://weather.af.mil/confluence/display/GALWEMHD/GALWEM+Transition+Help+Desk+Home.



```
<?xml version="1.0" encoding="UTF-8"?>
<RequestList xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="requestConfig.xsd">
        <Products>
                <Product id="GALWEM_0P25DEG"
fileName="galwemMeta_0P25DEG.xml" path="" selected="true"/>
                <Product id="GALWEM_17KM" fileName="galwemMeta_17KM.xml"
path="" selected="false"/>
        </Products>
        <RequestFormat format="GRIB2"/>
        <!-- NOTE:  if baseReferenceTime is null, treat it as LATEST -->
        <!--    <ForecastTime baseReferenceDate="" baseReferenceTime="" />  -->
        <Request>
                <Location>
                        <LatLongBox lowerLeftLatitude="35.0" lowerLeftLongitude="-
118.0" upperRightLatitude="45.0" upperRightLongitude="-106.0"/>
                </Location>
                <Time>
                        <ForecastTime baseReferenceDate="2019-07-16"
baseReferenceTime="12:00:00Z"/>
                        <ForecastPeriod>
                        <First>0</First>
                        <Last>3</Last>
                                <Increment>3</Increment>
                        </ForecastPeriod>
                </Time>
                <Parameter>
                        <File fileName="requestMyWIDA.xml" path=""/>
                </Parameter>
         </Request>
         <Request>
         …
         </Request>
</RequestList>
```

Fig. 2 Sample data-request definition file

3.5 Parameter and Level Definition File

In the previous data request definition file (Fig. 2), a file path is defined in the “File” element under the “Parameter” element. This file is used to specify the user’s desired parameters on particular level(s).

Figure 3 shows a portion of the parameter and level definition file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- 2D: Surface and AGL 2m/10m  -->
<urn:GriddedData xmlns:urn="urn:arl:wx:aois">
        <urn:GridParameter>
                <urn:Parameter urn:parameterName="pressure" />
                <urn:VerticalDimension>
                        <urn:MasterLayerName>SFC</urn:MasterLayerName>
                </urn:VerticalDimension>
        </urn:GridParameter>

        <urn:GridParameter>
                <urn:Parameter urn:parameterName="geopotentialHeight" />
                <urn:VerticalDimension>
                        <urn:MasterLayerName>SFC</urn:MasterLayerName>
                </urn:VerticalDimension>
        </urn:GridParameter>
        …

<!-- 2D Met Parameters, 2m or 10m above ground   -->
        <urn:GridParameter>
                <urn:Parameter urn:parameterName="temperatureAir" />
                <urn:VerticalDimension urn:upperLevel="2.0"
                        urn:lowerLevel="2.0">
                        <urn:MasterLayerName>TGL</urn:MasterLayerName>
                </urn:VerticalDimension>
        </urn:GridParameter>
        <urn:GridParameter>
                <urn:Parameter urn:parameterName="humidityRelative" />
                <urn:VerticalDimension urn:upperLevel="2.0"
                        urn:lowerLevel="2.0">
                        <urn:MasterLayerName>TGL</urn:MasterLayerName>
                </urn:VerticalDimension>
        </urn:GridParameter>
        …

<!-- 3D Met Parameters at pressure (Isobaric) levels, can specify levels  -->
        <urn:GridParameter>
                <urn:Parameter urn:parameterName="geopotentialHeight" />
           <urn:VerticalDimension urn:upperLevel="10" urn:lowerLevel="1013">

                        <urn:MasterLayerName>ISBL</urn:MasterLayerName>
                </urn:VerticalDimension>
        </urn:GridParameter>
        <urn:GridParameter>
                <urn:Parameter urn:parameterName="wind" />
           <urn:VerticalDimension urn:upperLevel="10" urn:lowerLevel="1013">

                        <urn:MasterLayerName>ISBL</urn:MasterLayerName>
                </urn:VerticalDimension>
        </urn:GridParameter>
        …
</urn:GriddedData>
```

Fig. 3 Sample parameter and level definition file

4. Multithread and Error Handling

Java's multithreading feature is used in this application for the M2M data transfer—one Java thread is created that corresponds to one GetCoverage Java object and multiple threads of the GetCoverage service call can be made simultaneously. The data returned by each thread execution (e.g., GetCoverage service call) are stored in a designated file. As a result, the M2M data transfer process is fast and efficient.

The failed service calls are recorded into a file. Users can inspect the contents of this file and re-execute those calls.

5. Application Deployment

This section discusses the installation, configuration, and execution of WCSClient.

5.1 Installation

The installation file "OGCClient_binary.zip" is created by the Apache Ant* script "build.xml".

To install the application, first extract the contents of "OGCClient_binary.zip" into a user-defined directory, then change ownership and have full permissions on all of the directories and files created.

5.2 Configuration

Logging level: Log4j is used for logging. By default, the logging level is set to INFO with the output to console. Edit log4j.properties as needed.

Java environment variable: Set up an environment variable Java so it points to the installed directory of Java SE Development Kit (JDK 1.8).

Property File "wcs.properties": Change the WCS URL keystore/truststore location and password as needed.

5.3 Execution

(A) Executables for GetCapabilities service:

- GetCapability.sh (UNIX)
- getCapability.bat (Windows)

*Apache Ant; c2019 [access 2019 Nov 11]. https://ant.apache.org/.

Usage: GetCapability.sh $dataPath

Where: $dataPath is the directory for returned information from service call.

(B) Executables for printing a list of coverageID for GALWEM model products:

- CoverageIDList.sh (UNIX)
- coverageIDList.bat (Windows)

Usage: CoverageIDList.sh $file

Where: $file is the XML-formatted file containing the information returned from GetCapability service call in (A).

(C) Executables for DescribeCoverage service:

- DescribeCoverage.sh (UNIX)
- describeCoverage.bat (Windows)

Usage: DescribeCoverage.sh $dataPath CoverageID

Where:

- $dataPath is the directory holding information returned from service call.
- CoverageID is the coverage ID (e.g., GALWEM_0P25DEG_201810-02-T06.00.00Z_ISBL). A list of the most recent coverage IDs for the GALWEM product can be obtained by executing (B).

(D) Executables for GetCoverage service:

- OrderData.sh (UNIX)
- orderData.bat (Windows)

Usage: OrderData.sh $dataPath

Where: $dataPath is the directory holding data returned from service calls and also files for failed orders.

Before executing the OrderData script, the user needs to edit the Data Request Definition File “requestConfig.xml” (Section 3.4) and Parameter and Level Definition File (Section 3.5) for the desired data request.

6. Conclusion

The WCSClient application consists of a set of Java packages, XML schemas, XML documents, property files, and UNIX/Windows scripts.

It provides the capability for M2M retrieval of NWP gridded data via WCS provided by the US Air Force's 557WW. Additionally, the application allows users to select a GALWEM model product and define data requests such as AOIs, forecast times, and parameters on particular level(s), and specifies a delivery data format.

7. References

1. The 557WW Open Geospatial Consortium (OGC) Services; c2019 [restricted website accessible to DOD personnel with a valid CAC]. https://weather.af.mil/confluence/display/SERVICES/OGC+Services.

2. The 557WW Web Coverage Service Documentation; c2019 [restricted website accessible to DOD personnel with a valid CAC]. https://weather.af.mil/confluence/display/SERVICES/Web+Coverage+Services+Documentation.

3. Java EE 6 Tutorial: RESTful Web Services; c2019 [accessed 2019 Nov 11]. http://docs.oracle.com/javaee/6/tutorial/doc/gijqy.html.

Appendix A. Sample Property File

```
########################################
# Property file wcs.properties for WCSClient,
# loaded from Java classpath
########################################

# WCS URL
webServiceEndPoint=https://gisweather.afwa.af.mil/services/WCS

# Define property "javax.net.ssl.keyStore" value
javax.net.ssl.keyStore=C:/Work/Project/common/etc/dod_certificate/keystore/mykeystore.jks
# if it is not set, System property defined by
# -Djavax.net.ssl.keyStore="..." will be used
#javax.net.ssl.keyStore=

# Define key store’s password
javax.net.ssl.keyStorePassword=changeit

# Defines property "javax.net.ssl.trustStore" value
javax.net.ssl.trustStore=C:/Work/Project/common/etc/dod_certificate/truststore/truststore.jks
# if it is not set, System property defined by
# -Djavax.net.ssl.trustStore="..." will be used
#javax.net.ssl.trustStore=

# Defines trust store’s password
javax.net.ssl.trustStorePassword=changeit

# NOTE: These values should not be changed
https.protocols=TLSv1,TLSv1.1,TLSv1.2,SSLv3
javax.net.ssl.trustStoreType=jks
javax.net.ssl.keyStoreType=jks
```

Appendix B. Shortened Version of Metadata for Forecast Model Product

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- 557WW GALWEM product 0.25 deg related properties -->
<galwemConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <modelid>GALWEM_0P25DEG</modelid>
    <!-- % is modelid, $ is level type -->
    <filenamePattern>%_YYYY-MM-DDTHH.00.00Z_$</filenamePattern>
    <boundedBy>
        <envelope srsName="CRS:84" axisLabels="Long Lat" uomLabels="deg deg"
            srsDimension="2">
            <lowerCorner>-180 -90</lowerCorner>
            <upperCorner>180 90</upperCorner>
        </envelope>
    </boundedBy>
    <timeFrame>
        <baseReferenceTime pattern="YYYY-MM-DDTHH.00.00Z">
            <hour>00 06 12 18</hour>
        </baseReferenceTime>
        <forecastTimeRange>
            <beginPosition>YYYY-MM-DDTHH:00:00Z</beginPosition>
            <endPosition>YYYY-MM-DD+6THH:00:00Z</endPosition>
        </forecastTimeRange>
    </timeFrame>
    <coverages>
        <coverage id="1" type="Ground">
            <parameters>
                <param name="albedo" pid="1" />
                 …
                <param name="weatherSnowFlag1Hr" pid="2" />
                …
                <param name="weatherRainFlag3Hr" pid="3" />
                 …

            <product pid="1" name="Ground_1"
                maskName="albedo_and_45_others">
                <forecastTime>
                    <first>0</first>
                    <last>240</last>
                    <posList>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 20  21 22 23 24 25 26 27 28 29 30  31 32 33 34 35 36 37 38 39 40 41 42 45 48 51 54 57 60 63
66 69 72 75 78 81 84 87 90 93 96 99 102 105 108 111 114 117 120 123 126 129 132 135 138
141 144 147 150 153 156 159 162 165 168 174 180 186 192 198 204 210 216 222 228 234
240</posList>
                </forecastTime>
            </product>
            <product pid="2" name="Ground_2"
                maskName="precipitationAccumulated1Hr_and_14_others">
                <forecastTime>
                    <first>1</first>
                    <last>42</last>
                    <posList>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42</posList>
                </forecastTime>
            </product>
```

```
                    …

              </coverage>

              <coverage id="2" type="AGL">
                    <parameters>
                     …
                    </parameters>
                    <product id=”1”…>
                    …
                   </product>
                   …
              </coverage>

              <coverage id="3" type="ISBL">
                    <parameters>
                          <param name="icingIntensity" pid="1" />
                          …
                          <param name="dustConcentration" pid="3" />
                          …
                    </parameters>

                    <product pid="1" name="ISBL_1"
maskName="divergenceRelative_and_1_other">
                          <forecastTime>
                                <first>0</first>
                                <last>240</last>
                                <!-- <increment>1</increment> -->
                                <posList>0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 45 48 51 54 57 60 63
66 69 72 75 78 81 84 87 90 93 96 99 102 105 108 111 114 117 120 123 126 129 132 135 138
141 144 147 150 153 156 159 162 165 168 174 180 186 192 198 204 210 216 222 228 234
240</posList>
                          </forecastTime>
                          <altitude unit="hPa">
                                <low>1013</low>
                                <high>10</high>
                                <levelList>1013 1000 975 950 925 900 850 800 750 700
650 600 550 500 450 400 350 300 250 200 150 100 70 50 30 20 10
                                </levelList>
                          </altitude>
                    </product>
                    …
                    <product pid="3" name="ISBL_3"
maskName="dustConcentration_and_1_other">
                    …
                    </product>
              </coverage>
              …
          </coverages>
</galwemConfig>
```

List of Symbols, Abbreviations, and Acronyms

557WW	557th Weather Wing
AOI	area of interest
ARL	US Army Research Laboratory
CAC	Common Access Card
DOD	US Department of Defense
GALWEM	Global Air–Land Weather Exploitation Model
GRIB1	GRIdded Binary or General Regularly-distributed Information in Binary, Edition 1
GRIB2	GRIdded Binary or General Regularly-distributed Information in Binary, Edition 2
ID	identification
Java	general-purpose computer-programming language
JAXB	Java Architecture for XML Binding
JDK	Java SE Development Kit
M2M	machine to machine
NWP	Numerical Weather Prediction
OGC	Open Geospatial Consortium
PKI	Public Key Infrastructure
REST	Representational State Transfer
UNIX	family of multitasking, multiuser computer operating systems
URL	Uniform Resource Locator
URI	Uniform Resource Identifier
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
XML	eXtensible Markup Language

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

1 CCDC ARL
(PDF) FCDD RLD CL
TECH LIB

1 CCDC ARL
(PDF) FCDD RLC ED
S ZENG