ARL-TN-0903 ● AUG 2018

US Army Research Laboratory

GRIB2 Data Decoder Application in MyWIDA: 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-0903 ● AUG 2018

US Army Research Laboratory

GRIB2 Data Decoder Application in MyWIDA: A Developer's Guide

by Subing Zeng
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
August 2018	Technical Note	October 2016–July 2018

4. TITLE AND SUBTITLE
GRIB2 Data Decoder Application in MyWIDA: 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)
US Army Research Laboratory
ATTN: RDRL-CIE-D
Battlefield Environment Division
Adelphi, MD 20783

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TN-0903

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 GRIB2 data decoder application provides the capability to process weather model data in Gridded Binary edition 2 format (GRIB2) and ingest data into the Gridded Meteorological database for use by MyWIDA. This report describes the design of the generic data decoder component of MyWIDA, then details the software dependency, implementation, and deployment of the GRIB2 data decoder application. It also outlines the GRIB2 data resources being incorporated by the application thus far.

15. SUBJECT TERMS
GRIB2, software design, implementation, deployment, weather model data, database, data decoder, terminology mapping

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Subing Zeng
Unclassified	Unclassified	Unclassified	UU	21	19b. TELEPHONE NUMBER (Include area code) 301-394-1961

Contents

List of Figures

List of Tables

1. Introduction

General Regularly-distributed Information in Binary form (GRIB) is a concise data format commonly used in meteorology to store historical and forecast weather data. The GRIB first edition (GRIB1)[1] is used operationally worldwide by most meteorological centers. But since GRIB second edition (GRIB2)[2] was introduced, more and more forecast model data are produced and distributed in this format.

MyWIDA is a rules-based tactical decision aid that provides weather effects on weapon systems by coupling forecast data with system-specific environmental rules.[3] The data decoder component of MyWIDA is responsible for processing weather model data from various data sources for use by MyWIDA. This component is designed to be generic, thus providing the framework for building a data decoder application for the weather model data in a specific data format.

The GRIB1 data decoder, which is used for processing weather model data in GRIB1, has been the sole data decoder application being developed and implemented in MyWIDA.

More recently, to utilize the weather model data in GRIB2 in MyWIDA, a new application named GRIB2 data decoder has been developed and implemented under the framework of the data decoder component.

This report describes the design, implementation, and operation of the GRIB2 data decoder. It is intended to help the developer understand the technical aspects of the application.

2. Design and Process Flow

2.1 Design of Generic Data Decoder

The design of the generic data decoder makes use of the factory pattern in the object-oriented program, providing an expendable and customized data processor framework for handling multiple weather model data formats.

Figure 1 depicts the instantiation of the specific data decoder. In the diagram, "ADataDecoder" represents an abstract base class from which concrete classes such as "Grib1Decoder" and "Grib2Decoder" are derived. "ADataDecoderFactory" represents an abstract factory class with a method "createDataDecoder" that decides which of those concrete decoders to return, based on the arguments passed to the method. For example, when GRIB2 is passed to the method "createDataDecoder", it returns an instance of the Grib2Decoder accordingly.

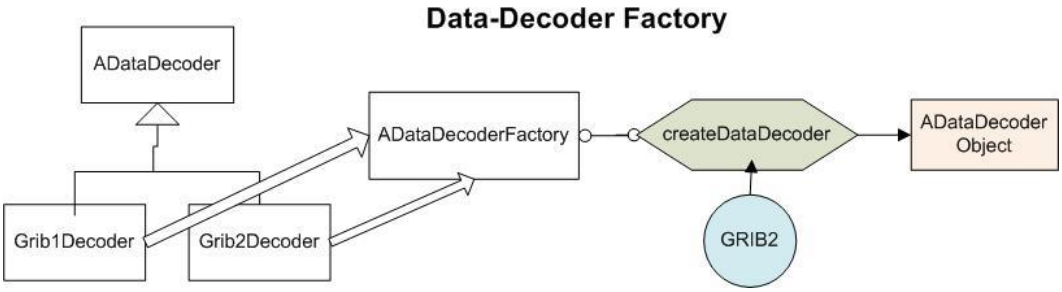

Fig. 1 Instantiation of data decoder

Table 1 shows a configuration file "datatype.properties", which is used to associate the data format with the specific Java class (i.e., specific data decoder application).

Table 1 Property file for managing data decoder

Filename: datatype.properties
Purpose: mapping a data format to a data decoder application
What to do: when a new decoder is implemented and the data format (e.g., GRIB2) is
defined in Java class "DataDecoderType", add a new row to this file:
data format is property, name of the Java class is property value
GRIB1=arl.grib1.Grib1Decoder
GRIB2=arl.grib2.Grib2Decoder

To create a new data decoder application for a specific data format under the framework of the data decoder component in MyWIDA, complete the following steps:

1) The data format that appears in the file (for example GRIB2) must be defined in the Java class DecoderDataType first:

   public final static DecoderDataType GRIB2= new DecoderDataType("GRIB2");

2) Create a concrete Java subclass of ADataDecoder (e.g., GRIB2DataDecoder) and provide the implementation for the method "decode", which is used for processing the weather model data with the data format just defined in Step 1.

3) Create an entry in the configuration file "datatype.properties". The property is the newly defined data format from Step 1 and the property value is the full path of Java class from Step 2.

4) Create UNIX/Windows scripts, which invoke this new data decoder application.

2.2 Process Flow

Figure 2 depicts the process flow of the data decoder application. First the method "createDataDecoder" from the "ADataDecoderFactory" class is invoked by the application. As a result, an instance of data decoder is created, based on the data format provided. Subsequently the method "decode" from this data decoder is invoked, then the data is decoded and ingested into a relational database.

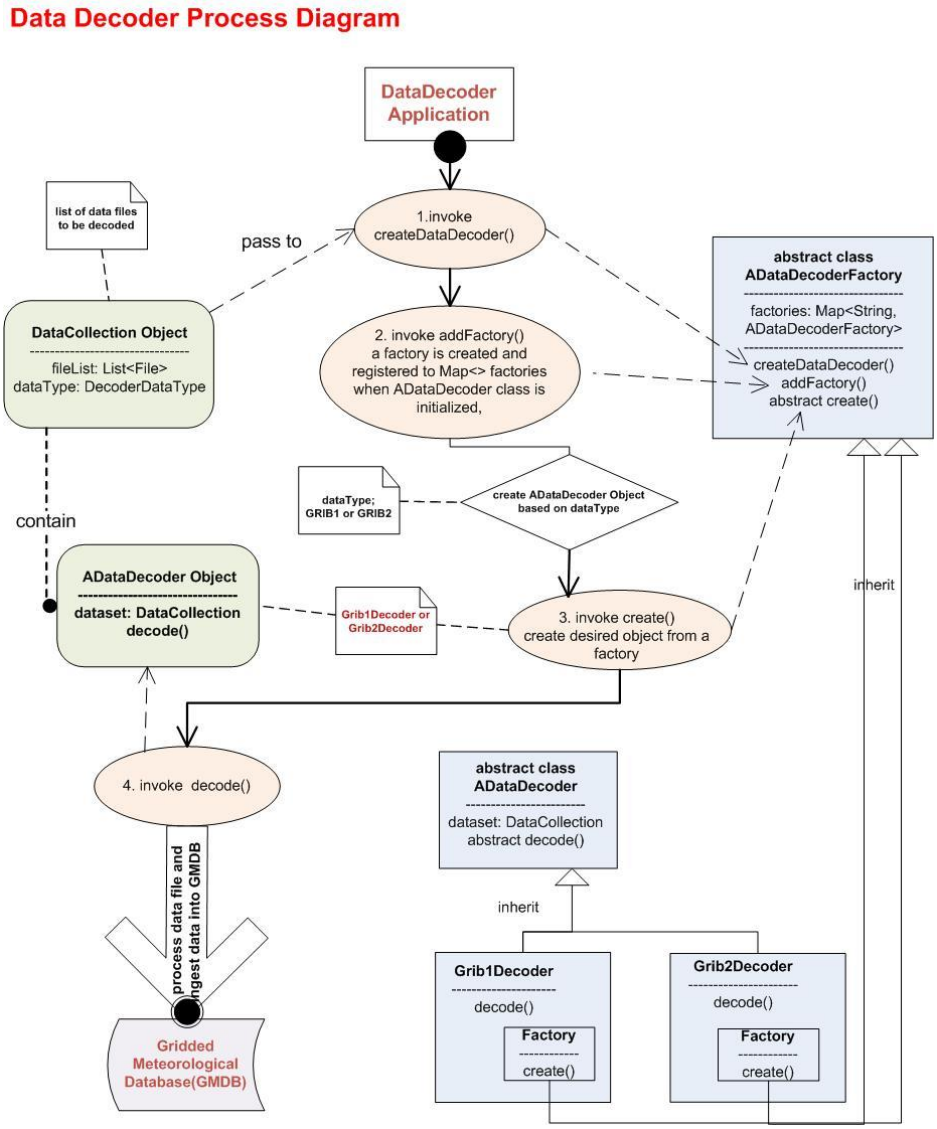

Fig. 2 Process flow of data decoder application

3. Dependency

The GRIB2 data decoder application is developed by applying three existing MyWIDA modules and some third party Java libraries.

3.1 In-House Module

Three modules developed in-house are utilized by the GRIB2 data decoder application:

- **The Gridded Meteorological Database (GMDB)**

  GMDB is a relational database module that models the atmosphere in four dimensions (4-D) with horizontal grids of weather parameter values at numerous vertical layers, over a number of forecast times for a specific base reference time. It is used for storing processed gridded meteorological model data and metadata.

- **Grid**

  This module provides the application programming interfaces (APIs) for making GMDB database connections, and retrieving and manipulating database records stored in the GMDB.

- **DataDecoder**

  This module includes the framework for implementing a data decoder application for a specific data format.

3.2 Third-Party Java Package

The following are Java packages from third-party that GRIB2 data decoder depends on:

- **General Java Packages**
  - commons-lang-2.1.jar
  - commons-io-1.4.jar
  - commons-cli-1.3.1.jar
  - log4j-1.2.17.jar
  - commons-logging-1.2.jar
- **NetCDF Java library**

The Network Common Data Form (NetCDF) Java library[4] by Unidata[5] is a Java interface to NetCDF files and other types of scientific data formats (including GRIB1/GRIB2). The API's design specifically for reading and decoding messages in GRIB2 format is utilized.

4. Implementation

The GRIB2 data decoder application consists of a set of Java packages, eXtensible Markup Language (XML) schemas, XML documents, and UNIX/Windows scripts. This section discusses the implementation aspects in detail.

4.1 Java Class and Package

Figure 3 shows the package and relationship within the data decoder component. The GRIB2 data decoder, as a subcomponent, is shown in the pink color block.

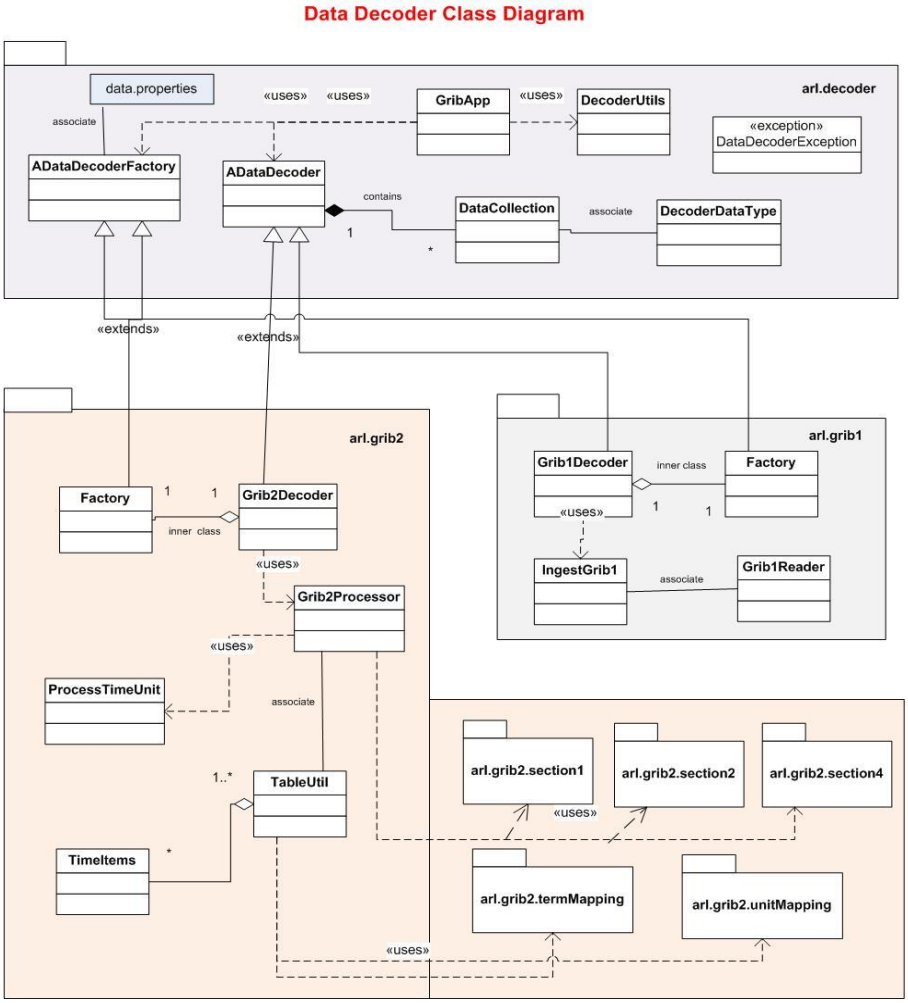

Fig. 3 Package and relationship of data decoder component

Table 2 shows the Java package and its main functions.

Table 2 Java package of GRIB2 data decoder

Package	Description
arl.decoder	Framework of data decoder
arl.grib2	Main package 1) For loading terminology/unit tables 2) For parsing tables to Java objects 3) For GRIB2 data processing
arl.grib2.section1	For mapping between Java objects and section 1 table in a GRIB2 message
arl.grib2.section3	For mapping between Java objects and section 3 table in a GRIB2 message
arl.grib2.section4	FOR mapping between Java objects and section 4 table in a GRIB2 message
arl.grib2.termMapping	Auto-generated classes by Java architecture for XML binding (JAXB), providing ability to match Java object to XML and the reverse for the terminology table
arl.grib2.unitMapping	Auto-generated classes by Java architecture for JAXB, providing ability to match Java object to XML and the reverse for the unit table
arl.grib2.ensemble	For processing ensemble GRIB2 data

4.2 Terminology Mapping

The terminology associations between MyWIDA and a specific GRIB2 data resource involves parameter mapping, level mapping, and unit mapping.

4.2.1 Parameter

The "termMapping.xsd" is an XML schema that defines the data structure for the parameter mapping between MyWIDA and the data resource.

To map the parameters defined by a specific data resource to the parameters defined in MyWIDA, an XML document that conforms to the above schema is created first. Inside this XML file, the parameters are grouped by the GRIB2 discipline, category, and local table value. The parameter from a particular data resource is placed in the <PARAM> element, the parameter used by MyWIDA is placed in the <GCPARAM> element. As a result a one-to-one relationship is established.

Figure 4 shows a shortened sample from the file "AFWA_57_0_14.xml", which represents the mapping of 557 Weather Wing (557WW) data resource (data center ID=57, sub-center ID=0, and GRIB2 table version=14) to MyWIDA.

```
<?xml version="1.0"?>
<TERMMAPPING xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
<!--  Table Information Section for AFWA/GALWEM -->
<TBLINFO tblname="AFWA_57_0_14.xml"
   gribedition="2"
   ctrid="57" subctrid="0"
   masterTableVerion="14"
   tableHigh="25"
   tableLow="0" />

<!--  Parameter Information Section, Mapping between PARAM and MyWIDA -->
 <DISCIPLINE id="0" name="Meteorology">
  <CATEGORY id="0" descrip="Temperature">
    <LOCALTABLE value="0" >
     <PARAM id="0" name="Temperature" units="K"  shortName="TMP">
       <GCPARAM gmdbID="11" gmdbName="temperatureAir"
            gmdbUnits="degreesKelvin"/>
     </PARAM>
     <PARAM id="2" name="Potential Temperature" units="K" shortName="POT">
      <GCPARAM gmdbID="254" gmdbName="temperaturePotential"
            gmdbUnits="degreesKelvin"/>
     </PARAM>
    …
   </LOCALTABLE>
  </CATEGORY>

  <CATEGORY id="1" descrip="Moisture">
 <LOCALTABLE value="0" >
      <PARAM id="0" name="Specific Humidity" units="kg kg-1" shortName="SPFH">
        <GCPARAM gmdbID="235" gmdbName="humiditySpecific"
                 gmdbUnits="kilogramsPerKilogram"/>
      </PARAM>
      <PARAM id="1" name="Relative Humidity" units="%" shortName="RH">
          <GCPARAM gmdbID="52" gmdbName="humidityRelative"
              gmdbUnits="percent"/>
      </PARAM>
       …
    </LOCALTABLE>
  </CATEGORY>
</DISCIPLINE>

<DISCIPLINE id="2" name="Land Surface">
   <CATEGORY id="0" descrip="Vegetation/Biomass">
    <LOCALTABLE value="0" >
      <PARAM id="0" name="Land Cover (0=sea, 1=land)" units="Proportion"
               shortName="LAND">
         <GCPARAM gmdbID="-1" gmdbName="N/A" gmdbUnits="N/A"/>
      </PARAM>
     …
   </LOCALTABLE>
  </CATEGORY>
</DISCIPLINE>
```

Fig. 4 A portion of file "AFWA_57_0_14.xml"

4.2.2 Unit

The “unitMapping.xsd” is an XML schema that defines the data structure for unit mapping and conversion. The “UnitMapping.xml” is an XML document that conforms to the above scheme and includes all the units supported by MyWIDA.

In the file “UnitMapping.xml”, the units are grouped by the GRIB2 discipline and category. The unit used in GRIB2 is placed in the <GRIB2UNIT> element, the unit used by MyWIDA is placed in the <GMDMUNIT> element. As a result a one-to-many relationship is established.

The unit conversion is achieved by the definition of attribute/value of “scale” and “offset” in the <GMDBUNIT> element.

Figure 5 shows the shortened sample of the file “UnitMapping.xml”.

```
<?xml version="1.0"?>
<UNITMAPPING xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" >
 <UNITTBLINFO tbldisName="UnitMapping.xml" gribedition="2" />
   <DISCIPLINE disId="0" disName="Meteorology" >
    <CATEGORY  catId="0" catName="Temperature"   >
       <GRIB2UNIT unitId="1" unitName="K">
            <GMDBUNIT id="1" symbol="degreesKelvin"  scale="1" offset="0"/>
            <GMDBUNIT id="2" symbol="degreesFahrenheit" scale="0.5555"
                  offset="255.3722"/>
            <GMDBUNIT  id="3" symbol="Celsius" scale="1" offset="273.15"/>
        </GRIB2UNIT>

         <GRIB2UNIT unitId="2" unitName="M m-2">
              <GMDBUNIT id="1"  symbol="wattsPermetersSquared" scale="1"
                   offset="0" />
         </GRIB2UNIT>
      </CATEGORY>

    <CATEGORY  catId="1" catName="Moisture">
       <GRIB2UNIT unitId="1" unitName="kg kg-1">
           <GMDBUNIT id="1"  symbol="kilogramsPerKilogram" scale="1" offset="0" />
       </GRIB2UNIT>
       <GRIB2UNIT unitId="2" unitName="%">
       <GRIB2UNIT unitId="5" unitName="PrecipitationType-NCEP"
             refTable="Table-4.201">
          <GMDBUNIT id="1"  symbol="coce"  />
       </GRIB2UNIT >

       <GRIB2UNIT unitId="6" unitName="CategoricalRain-NCEP"
            refTable="ncepTable-4.222">
          <GMDBUNIT id="1"  symbol="coce"   />
       </GRIB2UNIT >
     </CATEGORY>
 </DISCIPLINE >
  …
<UNITMAPPING>
```

Fig. 5 A portion of file “UnitMapping.xml”

4.2.3 Level

The level mapping between MyWIDA and GRIB2 is implemented by the method "processLayer" in the "Grib2Processor" Java class in the "arl.grib2" package.

4.3 Incorporation of Data Resource

The file "Grib2TableList.txt" (Table 3) is used to define the GRIB2 data resources currently being used by the GRIB2 data decoder.

Inside this file, a specific data resource is represented by a row. There are six columns in a row, and each column is separated by a semicolon. The column represents the data resource center ID, subcenter ID, GRIB2 master table version, generation process of the model, name of resource center, and name of XML terminology mapping file respectively.

Table 3 GRIB2 data resources

#center;	**subcenter;**	**master;**	**genProcess;**	**recourceName;**	**fileName**
57;	0;	4;	-1;	afwa;	AFWA_57_0_4.xml
7;	0;	2;	-1;	ncep;	NCEP_7_0_2.xml
57;	0;	14;	-1;	afwa;	AFWA_57_0_14.xml
7;	2;	2;	-1;	ncep;	NCEP_7_2_2.xml

The TableUtil.Java" is a Java class that contains methods for loading and parsing the unit and parameter mapping tables. It holds the references of the terminology mapping items between MyWIDA and multiple data resources.

To make a new GRIB2 data resource for use by MyWIDA, use the following steps:

1) Create an XML terminology mapping file for this new data resource.
2) Create a new row in the file "Grib2TableList.txt" that represents this new data resource.

Currently, three data resources have been incorporated into GRIB2 data decoder for use by MyWIDA. These three resources are the following:

- **The Global Air-Land Weather Exploitation Model (GALWEM)[6]**

  The GALWEM model data is produced and distributed by 557th WW.

- **The Global Forecast System (GFS)[7]**

  The GFS is a global numerical weather prediction computer model run by the National Oceanic and Atmospheric Administration/National Centers for Environmental Prediction (NOAA/NCEP).

- **The GFS Ensemble Forecast System (GEFS)[8]**

  The GEFS is a global ensemble numerical weather prediction computer model run by NOAA/NCEP.

5. Deployment

The GRIB2 data decoder is part of the data decoder component. The installation file "DataDecoder_binary.zip" is created by the Apache Ant [9] script "build.xml".

5.1 Installation

To install the data decoder component, first extract the content of "DataDecoder_binary.zip" into a user-defined directory, then change ownership and have full permissions on all the directories and files created.

5.2 Configuration

In order for the GRIB2 data decoder application to function properly, perform the following configuration after installation:

- **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.

- **GMDB**

  Edit database definition file "arldb.properties".

- **Java environment variable**

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

5.3 Execution

Executables for forecast model data are as follows:

- GribDecoder.sh (UNIX)

- gribDecoder.bat (Windows)

Usage

GribDecoder.sh $dataPath DataFormat

Where

$dataPath is the directory holding GRIB2 forecast data files.

DataFormat is the data format, e.g. GRIB1 or GRIB2.

Executables for ensemble forecast data are as follows:

- EnsembleDecoder.sh (UNIX)
- ensembleDecoder.bat (Windows)

Usage

EnsembleDecoder.sh $dataPath

Where

$dataPath is the location holding GRIB2 ensemble data files.

6. Summary and Conclusion

The GRIB2 data decoder application, developed under the framework of the data decoder component of MyWIDA, provides the capability to decode weather model data in GRIB2 format and ingest the data into the GMDB.

It consists of a set of Java packages, third-party Java libraries, configuration files, terminology and unit mapping files, build script, and executables.

Three GRIB2 data resources have been incorporated into the GRIB2 data decoder application for use by MyWIDA to date.

7. References

1. A guide to the code form FM 92-IX ext. GRIB edition 1. Geneva (Switzerland): World Meteorological Organization; 1994 May [accessed 2016 Aug 16]. Report No.: WMO-TD-611. http://www.wmo.int/pages/prog/www/WMOCodes/Guides/GRIB/GRIB1-Contents.html.

2. Guide to the WMO table driven code form used for the representation and exchange of regularly spaced data in binary form: FM 92 GRIB edition 2. Geneva (Switzerland): World Meteorological Organization; 2003 Jan 1 [accessed 2016 Aug 16]. https://www.wmo.int/pages/prog/www/WMOCodes/Guides/GRIB/GRIB2_062006.pdf.

3. Brandt J, Dawson L, Johnson J, Kirby S, Marlin D, Sauter D, Shirkey R, Swanson J, Szymber R, Zeng S. Second generation weather impacts decision aid applications and web services overview. White Sands Missile Range (NM): Army Research Laboratory (US); 2013 July. Report No.: ARL-TR-6525.

4. NetCDF Java Library. Unidata; c2018 [accessed 2018 Aug 16]. https://www.unidata.ucar.edu/software/thredds/current/netcdf-Java.

5. Unidata; c2018 [accessed 2018 Aug 16]. https://www.unidata.ucar.edu.

6. GALWEM. 557th Weather Wing [accessed 2018 Aug 16]. https://www.557weatherwing.af.mil.

7. Global Forecast System (GFS) Model. NCEP Central Operations. National Weather Service; 2014 Oct 17 [accessed 2016 Feb 10]. http://www.nco.ncep.noaa.gov/pmb/products/gfs.

8. Global Ensemble Forecast System (GEFS). NOAA National Centers for Environmental Information; 2012 July 12 [accessed 2017 Jan 1]. https://www.ncdc.noaa.gov/data-access/model-data/model-datasets/global-ensemble-forecast-system-gefs.

9. Apache Ant; c2018 [accessed 2018 Aug 16]. https://ant.apache.org.

List of Symbols, Abbreviations, and Acronyms

557WW	557th weather wing
API	application programming interface
ARL	US Army Research Laboratory
GALWEM	Global Air Land Weather Exploitation Model
GEFS	Global Ensemble Forecast System
GFS	Global Forecast System
GMDB	Gridded Meteorological Database
GRIB1	General Regularly-distributed Information in Binary form, edition 1
GRIB2	General Regularly-distributed Information in Binary form, edition 2
JAXB	Java architecture for XML binding
JDK	Java SE development kit
MyWIDA	My Weather Impacts Decision Aid
NCEP	National Centers for Environmental Prediction
NetCDF	network common data form
NOAA	National Oceanic and Atmospheric Administration
XML	eXtensible markup language

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIR ARL IMAL HRA RECORDS MGMT RDRL DCL TECH LIB
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
1 (PDF)	ARL RDRL CIE D S ZENG