OdimH5 User's Guide

Version 2.11
December 7, 2010

This document is a user's guide on how to use OdimH5.
It's still under development.

OdimH5 is a console utility designed to work on radar data files. It provides XML descriptor handler, HDF5 converter, Baltrad feeder mode.

1 Introduction

1.1 Overview

The OdimH5 is a Java-based tool working on meteorological radar data. It allows users to create an XML descriptor which contains all major information require to create a HDF5 file. The application also allows users to convert specific radar data to HDF5 format based on the descriptor. For `Baltrad` users it provides automatic online mode which feeds `BaltradDex` with actual data.

The application was implemented using the Java^{TM} 2 Platform, which is machine-independent.

HDF libraries

This release was built and tested with HDF5-1.8.4 Patch 1 with HDF5 1.6 compatibility flag. For information on new features in HDF5 Release 1.8.0 and format compatibility considerations, please visit
http://www.hdfgroup.org/HDF5/doc/ADGuide/CompatFormat180.html.

Platforms

This release was built and tested for the following platforms:

- Linux
- Linux x86_64
- Windows

1.2 Supported radar systems

This version can work with limited radar systems and products listed below.

Platforms

- Gematronik RAINBOW

Type of product

- Polar Volume Scan (ver. 5.2x and 5.3x)
  - dBZ - Reflectivity
  - uPhiDP - Differential Phase Shift
  - KDP - Specific Differential Phase Shift
  - RhoHV - Correlation Coefficient
- Cartesian image and composite (ver. 5.2x)
  - PPI - Plan Position Indicator
  - CAPPI - Constant Altitude PPI
  - MAX - Maximum Display
  - EHT - Echo Height
  - SRI - Surface Rainfall Intensity
  - PAC - Precipitation Accumulation
  - VIL - Vertical Integrated Liquid
  - HSHEAR - Horizontal Shear
- Vertical profile (Not implemented)
- Range-height indicator
  - RHI - Range Height Indicator

2 Getting Started

2.1 Installation

To get newest version of OdimH5 use `Git` a distributed revision control system and clone project from baltrad server. To do this use following command

```
git clone gitosis@git.baltrad.eu:OdimH5.git
```

or download it from Opera FTP server (`ftppro.knmi.nl`) and extract.

After downloading use `Apache Ant` to compile sources.

```
ant -Dprefix=/my/install/dir install
```

Administrator privileges might be needed

Program will be installed in `/my/install/dir`. If folder path parameter is not provided Odim will be installed in default folder `/opt/OdimH5`. Program uses HDF5 libraries, which are mostly included to the `.jar` file. However JNI interfaces files need separate installation.

In Linux: `libjhdf.so` and `libjhdf5.so` files which are provided with main program (`lib/linux`) must be included to the `LD_LIBRARY_PATH`.

In Windows: `jhdf.dll` and `jhdf5.dll` files which are provided with main program (`lib/win`) must be included to system environment variables.

2.2 Settings

The program reads options from `options.xml` file stored in the main folder. The following options can be provided:

`<radar>`	Every radar is represented by this element. The attribute `name` should be 3-letter radar name, same as one stored in raw volume file e.g. `name="SWI"`. This section contains one obligated tag: `WMO_id`, and following extra tags: `file_name`, `directory` and `nrays`
`<WMO_id>`	WMO block and station number
`<file_name>`	File name prefix compliant with ODIM file naming convention: `pflag_productidentifier_oflag_originator_`. For details see Opera document WD-2010-02.
`<directory>`	Path of the directory where radar volume data are stored. This directory will be watched by application, writing new files in the directory will awake the process of conversion.
`<nrays>`	Number of rays that are allowed, if volume contains more then that number, extra rays are discard.
`<baltrad>`	Section containing BALTRAD options for `server` and `sender`.
`<server>`	Address of HTTP server.
`<sender>`	Sender name.
`<ftp>`	Section containing FTP options for `address`, `login` and `password`.
`<address>`	Address of FTP server.
`<login>`	Login to FTP server.
`<password>`	Password to FTP server.

Example `options.xml` file:

```
<?xml version=\"1.0\" ?>
<!-- options -->
<options>
<radar name="BRZ">
<platform>RAINBOW</platform>
<WMO_id>12568</WMO_id>
<file_name>T_PAGZ46_C_SOWR_</file_name>
<directory>/home/volumes/BRZ_250_Z.vol</directory>
<nrays>360</nrays>
</radar>
<baltrad>
<server>http://172.30.9.34:8084/BaltradDex/transmitter.htm</server>
<sender>Baltrad.IMGW.pl</sender>
</baltrad>
<ftp>
<address>ftp.address</address>
<login>login</login>
<password>pass</password>
</ftp>
</options>
```

2.3 Convertion mode

`OdimH5` provides two conversion modes. First one creates XML descriptor, that can be used later to create HDF file. Second mode converts raw data directly to HDF5.

Prepare descriptor

Descriptor is an XML file, which structure corresponds to HDF5 file. To prepare descriptor use the following parameters:

`-i` Input file's path.
Program can work with only one file simultaneously.

`-o` Output file's path.
It is suggested to use `.xml` filename extension.

`-p` Radar platform's name.
At the moment only Gematronik's RAINBOW software is supported.

`-f` Product format.
Use one of the formats listed above according to input data type.

`-v` Verbose mode.
It is optional and displays status of progress of program work.

Example of use:

```
java -jar OdimH5.jar -i input.ppi -o ppi.xml -p RAINBOW -f IMAGE -v
```

Prepare HDF5 file from descriptor

It requires XML descriptor as an input file. To prepare HDF5 use the following parameters:

`-i` Input file's path.
Program can work with only one file simultaneously.

`-o` Output file's path.
It is suggested to use `.h5` filename extension. If no output file name provided program will generate one using ODIM convention, or if no prefix in options.xml were set program will use input file name with `.h5` filename extension.

`-v` Verbose mode.
It is optional and displays status of progress of program work.

Example of use:

```
java -jar OdimH5.jar -i ppi.xml -o output.h5
```

Prepare HDF5 file directly from raw file

It has the same parameters as descriptor preparation mode, but output file name has to end with `.h5`. For volumes `-o` parameter can be skipped, application will convert file directly to HDF5 using default name, which will be ODIM name when proper parameter will be provided in `options.xml` file or standard date-format name otherwise.

Examples of use:

```
java -jar OdimH5.jar -i input.ppi -o ppi.h5 -p RAINBOW -f IMAGE

java -jar OdimH5.jar -i input.vol -p RAINBOW -f PVOL
```

2.4 Baltrad Feeder

`OdimH5` allows users to send HDF5 files into `BaltradDex` system. To send a file use following command:

```
java -jar OdimH5.jar -i input.h5 -r Brzuchania -s IMGW.pl
-a http://172.30.9.34:8084/BaltradDex/dispatch.htm
```

It sends a single HDF5 file to the server but it can work as a continuous `Baltrad` feeder as well with online conversion to HDF5 format. It works automatically with specific options provided by user.

To run feeder use `-c` option and provide `Baltrad` details in `options.xml` in `<baltrad>` section.

Example of use:

```
java -jar OdimH5.jar -c
```

2.5 Sending file by FTP

`OdimH5` allows users to send HDF5 file using FTP. It works similar to `Baltrad` feeder. To run FTP feeder use `-c` option and provide `FTP` details in `options.xml` in `<ftp>` section.

2.6 Help

To display help menu in program use following parameter:

```
java -jar OdimH5.jar -h
```

3 Troubleshooting

Application is in its developing state and have not been tested thoroughly. To report a bug please send information to lukasz.wojtas@imgw.pl

4 Major Improvements and bug fixes

Version 2.6 (Release date: 2010-06-10)

- Added Baltrad feeder.
- Added direct HDF5 converter.

Version 2.11 (Release date: 2010-11-15)

- Sending files by FTP.
- RAINBOW 5.31.1 format supported.