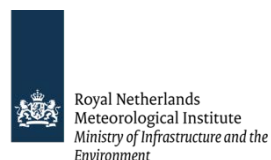**EUMETSAT**

NWP SAF

NUMERICAL WEATHER PREDICTION

Document NWPSAF-KN-UD-009
Version 2.1
February 2017

PenWP User Manual and Reference Guide

Anton Verhoef, Jur Vogelzang, Jeroen Verspeek and Ad Stoffelen

KNMI, De Bilt, the Netherlands

Met Office

ECMWF

Royal Netherlands
Meteorological Institute
Ministry of Infrastructure and the
Environment

METEO FRANCE
Toujours un temps d'avance

PenWP User Manual and Reference Guide

KNMI, De Bilt, the Netherlands

This documentation was developed within the context of the EUMETSAT Satellite Application Facility on Numerical Weather Prediction (NWP SAF), under the Cooperation Agreement dated 29 June, 2011, between EUMETSAT and the Met Office, UK, by one or more partners within the NWP SAF. The partners in the NWP SAF are the Met Office, ECMWF, KNMI and Météo France.

Change record			
Version	**Date**	**Author / changed by**	**Remarks**
1.9	May 2015	Anton Verhoef	First version for PenWP beta release
2.0	Oct 2015	Anton Verhoef	Version for first public PenWP release, split original UM into UM, PS and TLD docs
2.0.02	Nov 2015	Anton Verhoef	Modified according to DRI comments
2.1	Feb 2017	Jur Vogelzang	Added version 2.1 options

Contents

1 Introduction

This document is the User Manual (UM) of the Pencil beam Wind Processor (PenWP) software package. The current introductory section gives general information about PenWP. Section 2 provides information on how to install, compile, and link the PenWP software. The use of PenWP is discussed in section 3.

More information about PenWP can be found in several other documents. The Product Specification (PS) [1] contains more details about the design of PenWP. Reading the UM and the PS should provide sufficient information to the user who wants to apply PenWP as a black box.

The Top Level Design (TLD) of the code and the Module Design (MD) of the PenWP code can be found in [2].This document is of interest to developers and users who need more specific information on how the processing is done.

Please note that any questions or problems regarding the installation or use of PenWP can be addressed at the NWP SAF helpdesk at http://nwpsaf.eu/.

1.1 Aims and scope

The Pencil Beam Wind Processor (PenWP) is a software package written mainly in Fortran 90 with some parts in C for handling data from the SeaWinds (on QuikSCAT or ADEOS-II), OSCAT (on Oceansat-2 and ScatSat-1), HSCAT (on HY-2A) and RapidScat (on the International Space Station) scatterometer instruments, see table 1.1.. Details of these instruments can be found in [3] and [4], respectively, and on several web sites, see e.g. information on the NASA and ISRO web sites. PenWP is intended to be a generic wind processor for Ku-band pencil beam scatterometer data. It will be adapted to handle data from future instruments like the OSCAT successor ScatSAT (from ISRO) once they become available. Some background information on the development history of PenWP can be found in Appendix B.

Instrument	Satellite	Period of operations
SeaWinds	QuikSCAT	1999 – 2009
SeaWinds	ADEOS-II	2003
OSCAT	Oceansat-2	2009 – 2014
OSCAT	ScatSat	2016-
HSCAT	HY-2A	2011 -
RapidScat	ISS	2014 -

Table 1.1 Satellite instruments that can be processed with PenWP.

All Ku-band pencil beam scatterometers measure the ocean radar backscatter in very similar configurations and wavelengths and thus provide essentially the same geophysical measurements. After correction of the instrument gain calibration by ocean calibration and the performance of

verification tests, indeed standard processing of this category of instruments can be performed without loss of quality for any of the instruments. On the other side, the PenWP standard allows:

- A clearer separation of instrument and ocean response.

- Simplification of the processing of pencil-beam scatterometer data and further generic processing improvements.

- Ease of application.

PenWP generates surface winds based on pencil beam radar backscatter data. It has the ability to perform the ambiguity removal with the Two-dimensional Variational Ambiguity Removal (2DVAR) method and it supports the Multiple Solution Scheme (MSS). The output of PenWP consists of wind vectors which represent surface winds within the ground swath of the scatterometer. The input of PenWP is Normalized Radar Cross Section (NRCS), σ^0, data in BUFR format. These data may be near real-time. Conversion programs are included in the package to convert Hierarchical Data Format (HDF5) data from various instruments to BUFR. Output is written using the NOAA SeaWinds BUFR template by default or in the KNMI BUFR template with generic wind section. The NOAA BUFR template allows writing of up to 4 wind solutions whereas the KNMI BUFR template allows to write all 144 MSS wind solutions, see [1] for a description of these formats.

Apart from the scatterometer input data, PenWP needs Numerical Weather Prediction (NWP) model winds as a first guess for the Ambiguity Removal step. These data need to be provided in GRIB edition 1 or 2.

PenWP is developed as code which can be run in an operational setting as is done by, e.g., the Ocean and Sea Ice Satellite Application Facility (OSI SAF). The coding is mainly in Fortran 90 with some parts in C and has followed the procedures specified for the NWP SAF. Special attention has been paid to robustness and readability. PenWP may be run on every modern Unix or Linux machine. PenWP can also be run on a Windows machine if a Linux environment like the Windows Installer for Ubuntu (Wubi) is installed or on a MacOS machine (Darwin).

1.2   Testing PenWP

Modules are tested by test programs. Many test programs are part of the modules themselves. Test programs can be compiled separately. For PenWP, the description of the test programs and the results of the testing are reported in [5].

1.3　Why use PenWP?
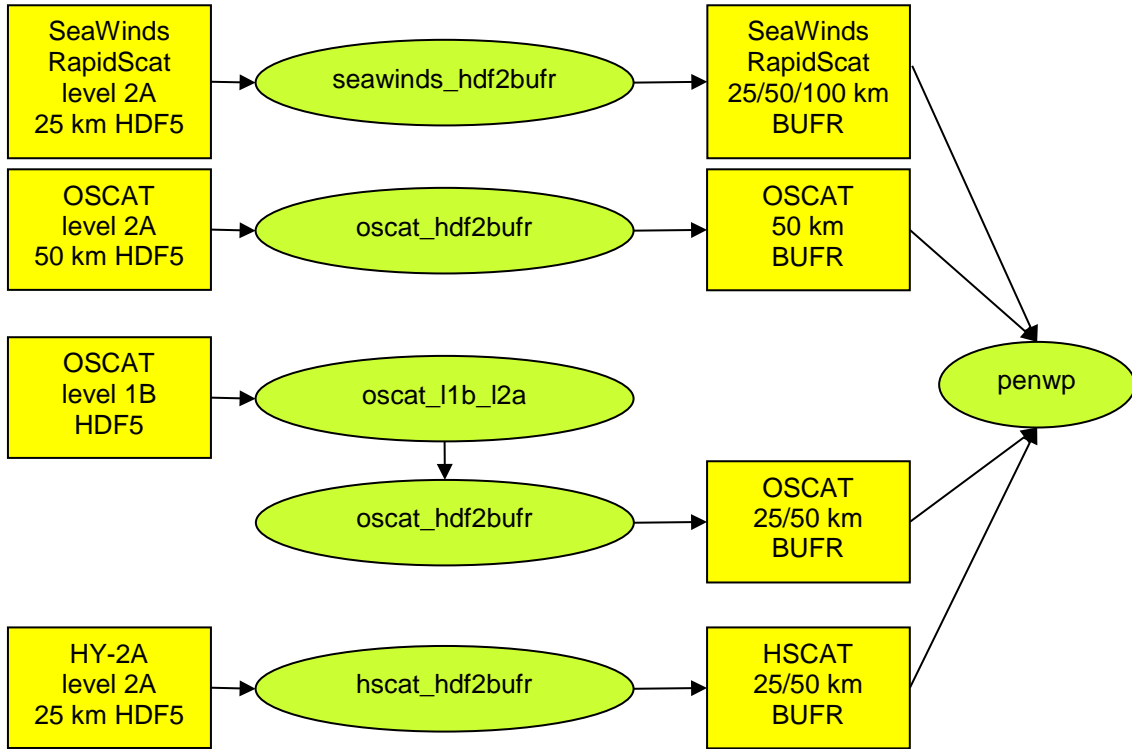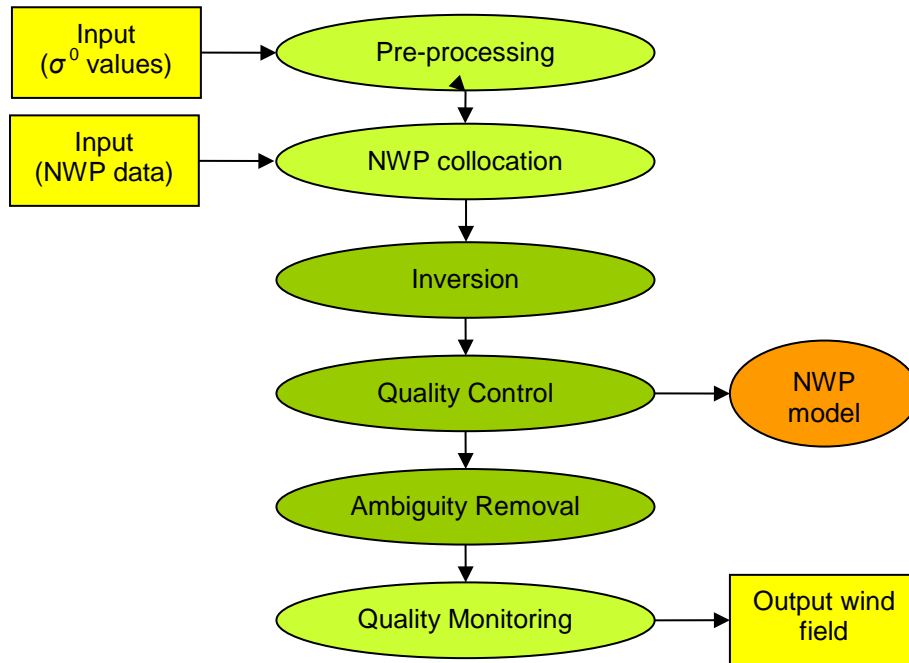

Figure 1.1　PenWP pre-processing scheme.

Scatterometers provide valuable observational data over the world's oceans. Therefore, successful assimilation of scatterometer data in numerical weather prediction systems generally improves weather forecasts. PenWP has been developed to fully exploit scatterometer data. It is meant to form the key component of the observation operator for surface winds in data assimilation systems.

The input of the PenWP main program is BUFR data which has been converted from HDF5 using one of the conversion programs (for SeaWinds/RapidScat, OSCAT or HY-2A) included in the package. The PenWP BUFR wind product can also be read and reprocessed. Figure 1.1 gives an overview of the different ways to obtain input BUFR data from scatterometer input data. Depending on the grid spacing of the input BUFR product, PenWP will process the data on 25 km, 50 km or 100 km grid spacing. The SeaWinds/RapidScat HDF5 to BUFR converter can create BUFR data on 25, 50 or 100 km grid spacing by averaging the backscatter data in the level 2a input file to the requested gridding. The OSCAT HDF5 to BUFR converter will create BUFR data with the same grid spacing as the level 2a input file. This can be 50 km when using the level 2a input data from the Indian Space Research Organisation (ISRO), or 25 km when using level 1b input data from ISRO in combination with a separate OSCAT level 1b to level 2a converter which is also included in the software package and which is based on software provided by NOAA. HY-2A input data are currently available on 25 km grid spacing but can also be averaged to a 50 km product.

The general scheme of PenWP (and any other wind scatterometer data processor) is given in figure 1.2. Besides scatterometer data, NWP model GRIB input containing land-sea mask, sea surface

temperature and first guess winds over the globe is necessary. NWP data is not needed when PenWP is used on level 2 BUFR data which already contains model winds and ice flagging, or when ambiguity removal is omitted. See the description of the command line options in section 3.1 for more information on this.

Figure 1.2   PenWP processing scheme. The wind vectors and their probabilities after Quality Control may be fed directly in the Data Assimilation step of a Numerical Weather Prediction model.

The PenWP processing chain contains several steps (see figure 1.2):

1. Pre-processing. The input file is decoded and the radar backscatter (σ^0) values are written in the data structures of PenWP. The individual backscatter measurements are averaged to a backscatter value on Wind Vector Cell level. Some quality control on the input data is done.

2. Collocation with NWP data. The GRIB edition 1 or 2 files containing NWP data are read and the values for land fraction, sea surface temperature and first guess winds are interpolated and stored with the information of each WVC.

3. Inversion. The σ^0 values are compared to the Geophysical Model Function (GMF) by means of a Maximum Likelihood Estimator (MLE). The wind vectors that give the best description of the σ^0 values (the solutions) are retained. The MLE is also used to assign a probability to each wind vector. The normal scheme allows 4 solutions at most, but in the Multiple Solution Scheme (MSS) the maximum number of solutions is 144.

4. Quality Control. Solutions that lie far away from the GMF are likely to be contaminated by, e.g., sea ice or confused sea state. During Quality Control these solutions are identified and flagged.

5. Ambiguity Removal. This procedure identifies the most probable solution using some form of external information. PenWP uses a two-dimensional variational scheme (2DVAR) as default. A cost function is minimized that consists of a background wind field and all solutions with

their probability, using meteorological balance, mass conservation and continuity as constraints.

6. Quality Monitoring. The last step is to output quality indicators to an ASCII monitoring file and to write the results in a BUFR format output file.

Steps 2 and 6 of the processing chain are rather trivial; the real work is done in steps 1, 3, 4, and 5.

As further detailed in [1], PenWP profits from developments in

- Inversion and output of the full probability density function of the vector wind (Multiple Solution Scheme, MSS) [6], [7].

- Quality Control (QC) [8], [9].

- Meteorologically balanced Ambiguity Removal (2DVAR) [10], with optional empirical background error correlations [11].

- Quality monitoring.

- Capability to process scatterometer data on 100 km, 50 km and 25 km cell spacing.

A complete specification of PenWP can be found in the Product Specification in [1]. The software is based on generic genscat routines. Genscat lies at the base of all scatterometer wind processors developed within NWP SAF. Genscat is a library of generic code suitable for all scatterometer instruments. It contains routines for e.g. wind inversion, ambiguity removal, reading and writing of files in BUFR, GRIB, and other formats. It also contains more low-level routines for e.g. date/time handling and numerical conversions. These routines are discussed in more detail in [2].

Genscat is owned by the NWP SAF. The code base is quite stable and most development work is done in the instrument-specific code that is built on top of genscat. For each new wind processor release, a dedicated genscat version is delivered which is tested by running the new configuration in parallel to the operational one and carefully comparing the outputs. Genscat is used as basis for all scatterometer wind data processors.

1.4 Modes of using PenWP

There are several modes to assimilate the scatterometer data in NWP models using PenWP. Anyway, the first thing to assure oneself of is the absence of biases by making scatter plots between scatterometer and NWP model first guess for at least wind speed, but wind direction and wind components would also be of interest to guarantee consistency; for more detailed guidance on bias correction see [11].

The scatterometer wind product, available as a deliverable from the EUMETSAT OSI SAF project, could be the starting point for NWP data assimilation:

1. The unique solution at every WVC may be assimilated as if it were buoy data. This is the fastest way and one exploits the data to a large extent. For a small advantage, PenWP could be installed to provide 2DVAR solutions based on the local first guess.

2. The PenWP software may be used to modify the 3DVAR or 4DVAR data assimilation system to work with the ambiguous wind solutions and their probabilities at every WVC in order to provide the full information content to the data assimilation system. This represents some investment, but the approach is generic and applicable to all scatterometer data. With respect

to option 1, this option only occasionally leads to an improved ambiguity removal, but often in dynamical atmospheric cases (storms or cyclones) that are really relevant.

Both options can be based on PenWP in standard or MSS mode [7]. MSS is somewhat more dependent on the balance constraints in 2DVAR or your own data assimilation system than the standard PenWP, but much less noisy. A substantial advantage is thus obtained by using option 2 and MSS, where potentially the full benefit of the scatterometer data is achieved. The mode of using PenWP thus depends on the opportunities, experience, and time the user has to experiment with scatterometer data in the NWP system under consideration; for more detailed guidance on scatterometer data assimilation see [11].

PenWP can, of course, also be used to create a stand-alone wind product, e.g., for nowcasting purposes. Such stand-alone wind products are deliverables of the OSI SAF project. More information on this project can be found on http://www.osi-saf.org/.

1.5   Conventions

Names of physical quantities (e.g., wind speed components u and v), modules (e.g. *BufrMod*), subroutines and identifiers are printed italic.

Names of directories and subdirectories (e.g. penwp/src), files (e.g. penwp.F90), and commands (e.g. penwp -f input) are printed in Courier. Software systems in general are addressed using the normal font (e.g. PenWP, genscat).

Hyperlinks are printed in blue and underlined (e.g. http://www.knmi.nl/scatterometer/).

2 Installing PenWP

The PenWP software package can be ordered from the NWP SAF website. A software request form is available on https://nwpsaf.eu/request_forms/index.html. PenWP is written in Fortran 90 (with a few low level modules in C) and is designed to run on a computer system under Linux or Unix. PenWP needs a Fortran 90 compiler and a C compiler for installation. PenWP comes along with a complete make system for compilation. In some cases, the Makefiles call installation scripts which are written in Bourne shell to enhance portability. When compiled, PenWP requires about 150-200 Mb disk space. The package is intended to be installed in a user directory (e.g. the home directory) and sufficient disk space and write permissions should be available. If the executable is to be installed into a system directory like `/usr/bin`, generally root permissions are necessary.

PenWP has been successfully implemented on many different computer platforms and with many different compilers and compiler versions. Hence we can not specify exact requirements to compiler versions. For example, PenWP has been tested recently with the following compiler versions:

- GCC version 4.8.3 of 20140911

- GNU Fortran (gfortran) version 4.8.3 20140911

- Portland Fortran compiler version 11.10-0

- G95 Fortran version GCC 4.0.3 (g95 0.94!) Jan 17 2013

In principle, PenWP may also run under Windows. However, it needs the BUFR and GRIB API libraries from ECMWF, and this poses some restrictions on the systems supported. Under Windows one must use a (free) Linux environment like Wubi (see http://www.ubuntu.com/download/desktop/windows-installer for more information and download).

To install PenWP, the following steps must be taken:

1. Copy the PenWP package (file `PenWP<version>.tar.gz`) to the directory from which PenWP will be applied, and unzip and untar it. This will create subdirectories `penwp` and `genscat` that contain all code needed (see section 2.1), and a script called `compile_penwp` for easy compilation.

2. Download the ECMWF BUFR library file `bufr_000405.tar.gz` (or another version not earlier than 000240) and copy it to directory `genscat/support/bufr`. Note that library versions 000388 and 000389 are not supported. See also section 2.2.

3. Download the ECMWF GRIB API library file `grib_api-1.14.0-Source.tar.gz` (or another version not earlier than 1.9.0) and copy it to directory `genscat/support/grib`. See also section 2.3.

8

4. Go to the top directory where the PenWP package (tar file) was copied to and run the `./compile_penwp` script. The script will ask for the compiler used and it will invoke the make system for compilation and linking of the software (see also section 2.5).

PenWP is now ready for use, provided that the environment variables discussed in section 2.6 have the proper settings. See also sections 2.6 and 3.1 for directions on how to run PenWP. The four steps in this section are sufficient to perform a standard compilation. Sections 2.2 to 2.5 provide more detailed information, especially for users interested to use advanced compiler options.

2.1   Directories and files

All code for PenWP is stored in a file named `PenWP<version>.tar.gz` that is made available in the framework of the NWP SAF project. This file should be placed in the directory from which PenWP is to be run. After unzipping (with `gunzip PenWP<version>.tar.gz`) and untarring (with `tar -xf PenWP<version>.tar`), the PenWP package is extracted in subdirectories `penwp` and `genscat`, which are located in the directory where the tar file was located. Subdirectories `penwp` and `genscat` each contain a number of files and subdirectories. A copy of the release notes can also be found in the directory `penwp/docs`.

Name	Contents
data	Look-up tables necessary in several processing steps
doc	Documentation, including this document
execs	Link to `penwp` executable, shell script for running PenWP
hscat	Tool to convert HSCAT HDF5 data to BUFR
oscat	Tool to convert OSCAT HDF5 data to BUFR
seawinds	Tool to convert SeaWinds and RapidScat HDF5 data to BUFR
src	Source code for PenWP program and supporting routines
tests	Example BUFR and GRIB input files for testing purposes.

Table 2.1 Contents of directory `penwp`.

Name	Contents
ambrem	Ambiguity removal routines
ambrem/twodvar	KNMI 2DVAR ambiguity removal routines
icemodel	Ice screening routines
inversion	Inversion and quality control routines
support	General purpose routines sorted in subdirectories
support/BFGS	Minimization routines needed in 2DVAR
support/bufr	BUFR tables (in subdirectory) and file handling routines
support/Compiler_Features	Compiler specific routines, mainly command line handling
support/convert	Conversion between wind speed/direction and *u* and *v*
support/datetime	Date and time conversion routines
support/ErrorHandler	Error handling routines
support/file	File handling routines
support/grib	GRIB file handling routines
support/hdf5	HDF5 handling routines
support/num	Numerical definitions and number handling routines

Name	Contents
`support/singletonfft`	FFT routines needed in minimization
`support/sort`	Sorting routines

Table 2.2 Contents of directory `genscat`.

Tables 2.1 and 2.2 list the contents of directories `penwp` and `genscat`, respectively, together with the main contents of the various parts. Depending on the distribution, more directories may be present, but these are of less importance to the user.

Directories `penwp` and `genscat` and their subdirectories contain various file types:

- Fortran 90 source code, recognizable by the `.F90` extension;

- C source code, recognizable by the `.c` extension;

- Files and scripts that are part of the make system for compilation like `Makefile_thisdir`, `Makefile`, `Objects.txt`, `use_`, and `Set_Makeoptions` (see 2.3.4 for more details);

- Scripts for the execution of PenWP in directory `penwp/execs`;

- Look-up tables and BUFR tables needed by PenWP;

- Files with information like `Readme.txt`.

After compilation, the subdirectories with the source code will also contain the object codes of the various modules and routines.

2.2   Installing the BUFR library

PenWP needs the ECMWF BUFR library for its input and output operations. Only ECMWF is allowed to distribute this software. It can be obtained free of charge from ECMWF at the web page https://software.ecmwf.int/wiki/display/BUFR/BUFRDC+Home. The package contains scripts for compilation and installation. The reader is referred to this site for assistance in downloading and installing the BUFR Library.

Directory `genscat/support/bufr` contains the shell script `make.bufr.lib`. It unzips, untars, and compiles the BUFR library file which is downloaded from ECMWF and placed into this directory. This script is part of the genscat make system and it is automatically invoked when compiling genscat. The current version is tested with BUFR version 000405, earlier versions as of 000240 can also be used. Note that library versions 000388 and 000389 are not supported.

BUFR file handling at the lowest level is difficult to achieve. Therefore some routines were coded in C. These routines are collected in library *bufrio* (see also [2]). Its source code is located in file `bufrio.c` in subdirectory `genscat/support/bufr`. Compilation is done within the genscat make system and requires no further action from the user (see 2.5).

2.3   Installing the GRIB API library

PenWP needs the ECMWF GRIB API library for its input operations. Only ECMWF is allowed to distribute this software. It can be obtained free of charge from ECMWF at the web page https://software.ecmwf.int/wiki/display/GRIB/Home. The package contains scripts for compilation and installation. The reader is referred to this site for assistance in downloading and installing the

GRIB API Library.

Directory `genscat/support/grib` contains the shell script `make.grib.lib`. It unzips, untars, and compiles the GRIB API library file which is downloaded from ECMWF and placed into this directory. This script is part of the genscat make system and it is automatically invoked when compiling genscat. The current version is tested with GRIB API version 1.14.0, but later versions (or earlier, but not earlier than 1.9.0) can be used. However, PenWP is not tested with later versions.

By default, the library for handling GRIB messages that are compressed using the JPEG algorithm, (as is done e.g. by NCEP) is not linked in the compilation process. This option can be activated by adding the link option `-lopenjpeg` to the `Makeoptions` file in directory `genscat`:

`LINKFLAGS = $(LIB) -lopenjpeg`

After this change, the software in directories `genscat/support/grib` and `penwp/src` needs to be recompiled using the commands `make clean` and `make`.

2.4   Installing the HDF5 library

The HDF5 software library from the HDF Group (http://www.hdfgroup.org/) is used by PenWP for reading and decoding HDF5 input files. See Appendix C for the copyright statement and the terms of use of this software. Binary libraries, compiled for different Linux and Unix platforms are delivered with PenWP in directory `genscat/support/hdf5/hdfgroup`. The `Makefile` in this directory tries to determine the operating system and creates a symbolic link from one of the binary libraries to a file called `libhdf5.a`. For example, directory `genscat/support/hdf5/hdfgroup` contains a library called `libhdf5_lin_i386.a` which is compiled for the 32 bits Linux platform. The `Makefile` will link this file to `libhdf5.a`, which in its turn will be linked when compiling PenWP. The same mechanism is used for some of the include files (`.h`) in this directory, which are also platform specific. This directory also contains the binary SZIP and ZLIB libraries that are used in conjunction with the HDF5 library.

Note that the collection of delivered libraries is by no means complete and it may be necessary for some platforms to download specific versions of the HDF5 software libraries from http://www.hdfgroup.org/ and to place them under the correct name in `genscat/support/hdf5/hdfgroup`. See the file `Readme.txt` in this directory for more information.

2.5   Manual compilation and linking

PenWP can be manually compiled (without using the `compile_penwp` script) when the user is interested to use more advanced compiler options. This is done in three steps:

1. Set the compiler environment variables according to the choice entered on request. This can be done by running the appropriate `use_*` scripts in directory `genscat`.

2. Go to directory `genscat` and type `make`.

3. Go to directory `penwp` and type `make` to produce the executable `penwp` in directory `penwp/src`.

Before activating the make system, some environment variables identifying the compiler should be set. These variables are listed in table 2.4. The environment variables in table 2.4 can be set by using one of the `use_*` scripts located in directory `genscat`. Table 2.5 shows the properties of these scripts. The scripts are available in Bourne shell (extension `.bsh`) and in C shell (extension `.csh`). Note that if one of the environment variables is not set, the default `f90` and `cc` commands on the Unix platform will be invoked. Note that in the top directory a script called `compile_penwp` is provided that asks the user which compiler he wants to use and invokes the appropriate `use_*` script (step 1 above), after which the compilation in the `genscat` and `penwp` directories is performed (steps 2 and 3 above).

Variable	Function
$GENSCAT_F77	Reference to Fortran 77 compiler
$GENSCAT_F90	Reference to Fortran 90 compiler
$GENSCAT_CC	Reference to C compiler
$GENSCAT_LINK	Reference to linker for Fortran objects
$GENSCAT_CLINK	Reference to linker for C objects
$GENSCAT_SHLINK	Reference to linker for shared objects

Table 2.4 Environment variables for compilation and linking.

Script	Fortran compiler	C compiler	Remarks
use_g95	g95	gcc	GNU compilers by Andy Vaught
use_gfortran	gfortran	gcc	GNU-GCC compiler collection
use_ifort	ifort	icc	Intel Fortran and C compilers
use_pgf90	pgf90	gcc	Portland Fortran compiler

Table 2.5 Properties of the use_* scripts.

Example: To select the GNU g95 compiler under Bourne, Bash or Korn shell type ". use_g95.bsh", the dot being absolutely necessary in order to apply the compiler selection to the current shell. Under C shell the equivalent command reads "`source use_g95.csh`".

If the user wants to use a Fortran or C compiler not included in table 2.6, he can make his own version of the `use_*` script, or set the environment variables for compilation and linking manually.

PenWP is delivered with a complete make system for compilation and linking under Unix or Linux. The make system is designed as portable as possible, and system dependent features are avoided. As a consequence, some tasks must be transferred to shell scripts. The make system consists of two parts: one for PenWP and one for genscat. The genscat part should be run first. For compilation and linking of the genscat part, the user should move to the `genscat` directory and simply enter `make`.

The `Makefile` refers to each subdirectory of genscat, invoking execution of the local `Makefile` and, in cases where a subdirectory contains code as well as a subdirectory containing code, `Makefile_thisdir`. The settings for the compilers are located in file `Makeoptions` in directory `genscat`. This file is generated by the Bourne shell script `Set_Makeoptions`

which is called automatically by the genscat make system. The local `Makefile` in subdirectory `genscat/support/bufr` calls the script `make.bufr.lib` for compilation of the BUFR library (see 2.2). It also contains the Fortran program `test_modules` that generates the binary BUFR tables B and D from the ASCII tables already present, and is executed automatically by the make system. Program `test_modules` can also be used to test the genscat BUFR module. The `Makefile` in subdirectory `genscat/support/bufr/bufr_tables` calls some shell scripts, which make symbolic links (using the `ln -s` command) of the generic BUFR tables B and D under different names. There are four different naming conventions in BUFR version 000240 to 000280, and binary files are generated for each of them. Further information on the make system is given in the inline comments in the scripts and Makefiles.

Compilation and linking of the PenWP part is done in a similar manner: go to the `penwp` directory and enter `make`. As with genscat, the make system will execute Makefiles in every subdirectory of `penwp`. The result is the executable `penwp` in directory `penwp/src` and a symbolic link to this executable in `penwp/execs`. PenWP is now ready for use. The make system of PenWP doesn't need any further files except the genscat file `Makeoptions`. This is the reason why genscat should be compiled first.

When recompiling (part of) PenWP or genscat with the make system, for instance when installing a new version of the BUFR library, one should be sure to enter `make clean` first. To recompile part of the software invoke the make system where needed. The compiler settings from file `Makeoptions` in directory `genscat` will be used again. If a change in these settings is necessary, type `make clean` in the genscat directory and `Makeoptions` will be removed. Don't forget to rerun the `use_*` commands to select the right compiler.

2.6   Environment variables

PenWP needs a number of environment variables to be set. These are listed in table 2.3 together with their possible values.

Name	Value
$BUFR_TABLES	genscat/support/bufr/bufr_tables/
$GRIB_DEFINITION_PATH	genscat/support/grib/definitions
$LUT_FILENAME_KU_HH	penwp/data/\<platform>/nscat4_250_73_51_hh.dat
$LUT_FILENAME_KU_VV	penwp/data/\<platform>/nscat4_250_73_51_vv.dat
$LUTSDIR	penwp/data
$NBEC_FILE	genscat/ambrem/nbec_ascat-a-coa_cos-auto-4000_tccal_obserrcorr.dat

Table 2.3 Environment variables for PenWP.

The `$BUFR_TABLES` variable guides PenWP to the BUFR tables needed to read the input and write the output. The `$GRIB_DEFINITION_PATH` variable is necessary for a proper functioning of the GRIB decoding software.

The variables `$LUT_FILENAME_KU_HH` and `$LUT_FILENAME_KU_VV` point PenWP to the correct binary Ku band GMF lookup tables (LUT) at HH and VV polarisation, respectively. They should contain a file name including a valid path. NSCAT lookup tables are delivered with PenWP in big endian and little endian binary formats, the `<platform>` part in the paths should be set to

`big_endian` or `little_endian` depending on your computer platform type.

The variable $LUTSDIR points PenWP to a directory containing some look up tables that are used to normalise the inversion residuals and to compute atmospheric attenuations for the Ku band radar data. The necessary tables are delivered with PenWP.

The variable $NBEC_FILE points PenWP to the file from which it can read the Numerical Background Error Correlations, to be used optionally in 2DVAR.

It may happen that running PenWP results in an error message like 'error while loading shared libraries: libzzz'. In such cases it may help to locate the specified shared object file on your system and to set the $LD_LIBRARY_PATH variable to this directory.

3 Processing details

3.1 Command line options

The PenWP main program is started from directory `penwp/execs` with the command

`penwp [options/modes] -f <input file> [-nwpfl <file>]`

with `<>` indicating obligatory input, and `[]` indicating non-obligatory input. For normal PenWP operation we recommend to use the `penwp/execs/penwp_run` script which will call PenWP and pass its command line options to the PenWP executable. The following command line options are available.

`-f <input file>` Process a BUFR input file with name `input file`.

PenWP can only handle data in BUFR format. Conversion tools are available to convert HDF level 2a or level 2b data from SeaWinds, RapidScat, OSCAT or HSCAT to BUFR, see below in this section.

Example: `penwp -f QS_D08001_S0152_E0323_B4444445` will process this file. The results will be written to a new BUFR file, see below in this section for the output file naming convention. It is possible to concatenate multiple BUFR input files into one using the Unix `cat` command.

`-nwpfl <file>` Read a list of GRIB file names in the file named `file`.

The files in the list are read and the GRIB edition 1 or 2 data are used in the wind processing. The most convenient way to construct a file list is to use the Unix command `ls -1 GRIB file pattern > file`. Note that the '-1' contains the number '1' and not the character 'l'. If no GRIB data are used, only the land masking which is present in the level 2a/b files will be used. No ice screening will be performed (unless the `-icemodel` option is used). Ambiguity removal will be performed only if model winds are already present in the input BUFR file (i.e., in case of reprocessing of a level 2 file) or if the `-armeth 1strank` option is used (i.e., selection of the 1st rank wind solution). If level 2 data are reprocessed and no NWP data are read, the `qual_sigma0` flag which was set in the initial processing is evaluated and it will be used to determine if a WVC contains suitable backscatter data for wind inversion.

Several options for the processing can be invoked.

`-noinv` Switch off inversion (default is switched on).

This option is useful to save processing time when wind inversion is already done in the input file, i.e., when a PenWP BUFR output file is reprocessed. For example, the NWP background winds can be replaced

15

and the ambiguity removal performed in one run without repeating the wind inversion.

-icemodel   Switch on ice screening (default is switched off).

When switched off, no ice screening is done, except when a GRIB file containing sea surface temperature is read. The command line option invokes the Bayesian ice model which keeps the history of each location and uses this history to determine the sea or ice state of a WVC. The ice screening can be combined with the ice screening which is done in the GRIB collocation. In this case, the SST of the GRIB file will be used to assign a WVC as 'surely water' when the SST is above+5.0 °C, to support the Bayesian ice screening.

-noamb   Switch off ambiguity removal (default is switched on).

This option is useful if the selection of the scatterometer wind solution is left to the data assimilation procedure of a Numerical Weather Prediction model. In other words: the NWP model is fed with a number of solutions and their probability, and finds the best value when comparing with other data sources.

-nowrite   Do not produce BUFR output (default is switched on).

-noc   Perform σ^0 calibration (NWP Ocean Calibration, default is switched off).
A calibration of the σ^0. values is performed, i.e., the backscatter values are changed in order to obtain better calibrated winds.

-mss   Use the Multiple Solution Scheme for Ambiguity Removal.

If the Multiple Solution Scheme (MSS) is switched on, PenWP internally works with 144 different solutions for the wind vector. If MSS is switched off, PenWP calculates four solutions at most. MSS is switched off as default. Note that in the NOAA BUFR format, only 4 out of these 144 wind solutions will be written. The KNMI BUFR format may contain up to 144 wind solutions, see the -genericws option.

-nbec   Use Numerical Background Error Correlations (NBECs) in 2DVAR instead of the default Gaussian error correlations. NBECs give more weight to the observations, resulting in better agreement with buoy winds and less frequent setting of the KNMI QC and VarQC flags at the cost of increased processing time.

-armeth <meth>   Choose ambiguity removal method.
Valid methods are: 1strank - the wind solution with the lowest distance to the GMF (residual) is selected, bgclosest - the wind solution closest to the background model wind is selected, 2dvar - 2DVAR, see [2]. The default is 2dvar.

-genericws <N>   This option generates a second BUFR output file in the KNMI generic wind section format.
The number of wind solutions <N> to be written into the KNMI BUFR

format is flexible due to the use of the so-called delayed replication and can be chosen between 1 (providing only the selected wind solution) and 144 (providing all wind solutions in MSS processing). The KNMI BUFR format is not approved by the WMO but can be handled by generic decoders.

-binof <file>   Write selected data of each WVC to a binary output file.

Data are written to a binary file `<file>`. This option is intended for research activities. More information on the file format can be found in the Fortran code of PenWP.

-mon   Switch on the monitoring function.

The monitoring results are written in an ASCII file with the name `<name of BUFR output file>.mon`. By default, no monitoring file is produced.

-verbosity <L>   Set the verbosity level to L (default is 0).

If the verbosity level is -1 or smaller, no output is written to the standard output except error messages. If the verbosity level equals 0 only some top level processing information is written to output. If the verbosity level is 1 or greater, also additional information is given.

Further, there is an option `-par` to change the 2DVAR settings, but this option is for internal research purposes and is not supported for the public releases.

The normal mode of operation of PenWP is wind processing, i.e., a BUFR file is read and the various processing steps are performed.

Besides the wind processing, some other modes of operation are available. If one of the modes is invoked, PenWP internally sets some of the options in order to obtain the desired result. Note that these modes are always used in combination with the `-f <input file>` option.

-mononly   Write the monitoring file without any processing.

-properties   Write some properties of the last row of the input file.

The data acquisition date and time of the last row are written to a small ASCII output file `properties.txt`.

-writeonly   Write all data to BUFR output without processing.

This mode is useful to copy an input file to BUFR output without processing.

Running the command `penwp` without any command line options will display a list of all available command line options with a short explanation on the console. Running the command `penwp` with an illegal option will produce the same output, but preceded by an error message.

The output will be written into a BUFR file with the name

`INSTR_YYYYMMDD_HHMMSS_SAT_ORBIT_srv_o_SMPL_ovw_l2.bufr`,

where

- `INSTR` is the instrument, `seawi`, `oscat`, `hscat` or `rapid`.

- `YYYYMMDD_HHMMSS` is the acquisition date and time (UTC) of the first data in the file.

- `SAT` is the satellite (6 characters), `qscat_`, `adeos2`, `ocsat2`, `scasa1`, `hy2a_` or `iss___`.

- `ORBIT` is the orbit number (5 digits) of the first data in the file.

- `SMPL` is the WVC sampling (cell spacing), `1000` for 100 km, `500` for 50 km and `250` for 25 km.

- If the above convention results in identical input and output file names, the extension '~' is added to the output file name.

- If the option `-genericws` (see above) is used, a second BUFR file with the same name and extension '`.genws`' is written.

Example: the output file name can look like:
`oscat_20111107_135116_ocsat2_11243_o_500_ovw_l2.bufr`
(Oceansat-2 50 km) or
`seawi_20080107_114220_qscat__44536_o_1000_ovw_l2.bufr`
(QuikSCAT 100 km)

3.2   Scripts

Directory `penwp/execs` contains a Bourne shell script `penwp_run` for running penwp with the correct environment variables. The script can be invoked with all valid command line options for penwp.

3.3   HDF conversion tools

Several tools are available to convert HDF data from SeaWinds/RapidScat, OSCAT and HSCAT to BUFR.

Directory `penwp/seawinds` contains the executable `seawinds_hdf2bufr`. It can read SeaWinds or RapidScat level 2a and level 2b files in HDF and will write the data in BUFR format. For level 2a, only the backscatter information in the BUFR output will be filled; for level 2b, only the wind information in the BUFR output will be filled. In the latter case wind processing is not possible since the HDF level 2b product does not contain σ^0 data but only winds. The backscatter 'eggs' in the level 2a data are accumulated to WVC level backscatter data, see[1]. Note that the SeaWinds data need to be converted from HDF4 to HDF5 before running `seawinds_hdf2bufr`. This can be done using a tool called h4toh5 which is available from the HDF Group: http://www.hdfgroup.org/h4toh5/:

h4toh5 QS_S2A44444.20080021532 QS_S2A44444.20080021532.h5

The resulting file can be used as input for `seawinds_hdf2bufr`:

seawinds_hdf2bufr [options] -f <HDF5 file> -o <BUFR file>

to create one output file containing all data, or:

seawinds_hdf2bufr [options] -f <HDF5 file> -wpo

to create one output file per orbit. The available options are:

-properties   Write some properties of the last row of the input file.
                      The acquisition date and time are written to a small ASCII output file `properties.txt`.

-stride 2|4   Accumulate 25 km level 2a data with a stride of 2 or 4 to a 50 km or 100 km BUFR product. The 100 km product is not available for RapidScat.

Directory `penwp/oscat` contains the executable `oscat_hdf2bufr`. It can read OSCAT level 2a and level 2b data from ISRO and will write the data in BUFR format. For level 2a, only the backscatter information in the BUFR output will be filled; for level 2b, only the wind information in the BUFR output will be filled. In the latter case wind processing is not possible since the level 2b product does not contain σ^0 data. The conversion can be done with:

`oscat_hdf2bufr [options] -f <HDF5 file> -o <BUFR file>`

The available options are:

-allswath   Distribute outer swath (VV-polarised) slices over four WVC beams.
                     This option will re-distribute the VV slice information over four (more or less independent) backscatter data. The VV fore slice data will be split into two sets and the VV aft slice data as well. Slices are accumulated and averaged based on their azimuth angles, see [1].

-properties   Write some properties of the last row of the input file.
                     The acquisition date and time are written to a small ASCII output file `properties.txt`.

Directory `penwp/oscat/l1b_l2a` contains the executable `oscat_l1b_l2a`. It can read OSCAT level 1b data from ISRO and will write a level 2a HDF5 file with 25 km or 50 km WVC spacing, which can be further processed with oscat_hdf2bufr and PenWP. The conversion can be done with:

`oscat_l1b_l2a [options] -f <L1B file >`

The available options are:

-d   Output directory where the level 2a file is written. By default, the current directory is used.

-s   WVC spacing of level 2a output in km. Allowed values are 25 and 50.

Directory `penwp/hscat` contains the executable `hscat_hdf2bufr`. It can read HSCAT level 2a and level 2b data from NSOAS and will write the data in BUFR format. For level 2a, only the backscatter information in the BUFR output will be filled; for level 2b, only the wind information in the BUFR output will be filled. In the latter case wind processing is not possible since the level 2b product does not contain σ^0 data. The conversion can be done with:

`hscat_hdf2bufr [options] -f <HDF5 file> -o <BUFR file>`

The available options are:

-allswath   Distribute outer swath (VV-polarised) slices over four WVC beams.
                     This option will re-distribute the VV slice information over four (more or

less independent) backscatter data. The VV fore slice data will be split into two sets and the VV aft slice data as well. Slices are accumulated and averaged based on their azimuth angles, see [1].

-properties   Write some properties of the last row of the input file.
                   The acquisition date and time are written to a small ASCII output file `properties.txt`.

-stride 2   Accumulate 25 km level 2a data with a stride of 2 to a 50 km BUFR product.

3.4   Test data and test programs

Directory `penwp/tests` contains one HDF5 file for testing the PenWP executable. File `S1L2A2012006_12113_12114_2.h5.gz` contains (gzipped) OSCAT level 2a data from 6 January 2012, 13:51 to 14:03 UTC with 50 km cell spacing, as obtained from ISRO. The files `ECMWF*.grib` contain the necessary NWP data (SST, land-sea mask and wind forecasts) to perform the NWP collocation step.

The user can test the proper functioning of PenWP using the files in the `penwp/tests` directory. To do this, first create a small file containing a list of NWP files:

```
ls -1 ECMWF_* > nwpflist
```

Note that the '-1' contains the number '1' and not the character 'l'. Then, gunzip the HDF5 file:

```
gunzip -c S1L2A2012006_12113_12114_2.h5.gz >
S1L2A2012006_12113_12114_2.h5
```

Set the $BUFR_TABLES environment variable:

```
export BUFR_TABLES=../../genscat/support/bufr/bufr_tables/
```

for Korn shell or Bourne shell or

```
setenv BUFR_TABLES ../../genscat/support/bufr/bufr_tables/
```

for C shell. Convert the level 2a input file to BUFR:

```
../execs/oscat_hdf2bufr -f S1L2A2012006_12113_12114_2.h5 -o
oscat.bufr
```

Then run PenWP:

```
../execs/penwp_run -f oscat.bufr -nwpfl nwpflist -mss -mon -noc
```

The result should be an OSCAT level 2 file in BUFR format, called `oscat_20120106_135109_ocsat2_12113_o_500_ovw_l2.bufr`.
Figure 1.3 shows the global coverage of the test run. The colours indicate the magnitude of the wind speed as indicated by the legend.

Note that conversion software is delivered with PenWP to convert the BUFR output files into ASCII or NetCDF formats which are more convenient to visualise using publicly available software tools. These tools can be found in in directories `genscat/tools/bufr2asc` and `genscat/tools/bufr2nc` and are ready to use after compilation of PenWP. More

information on the use of this software is available on http://www.knmi.nl/scatterometer/bufr_reader/.
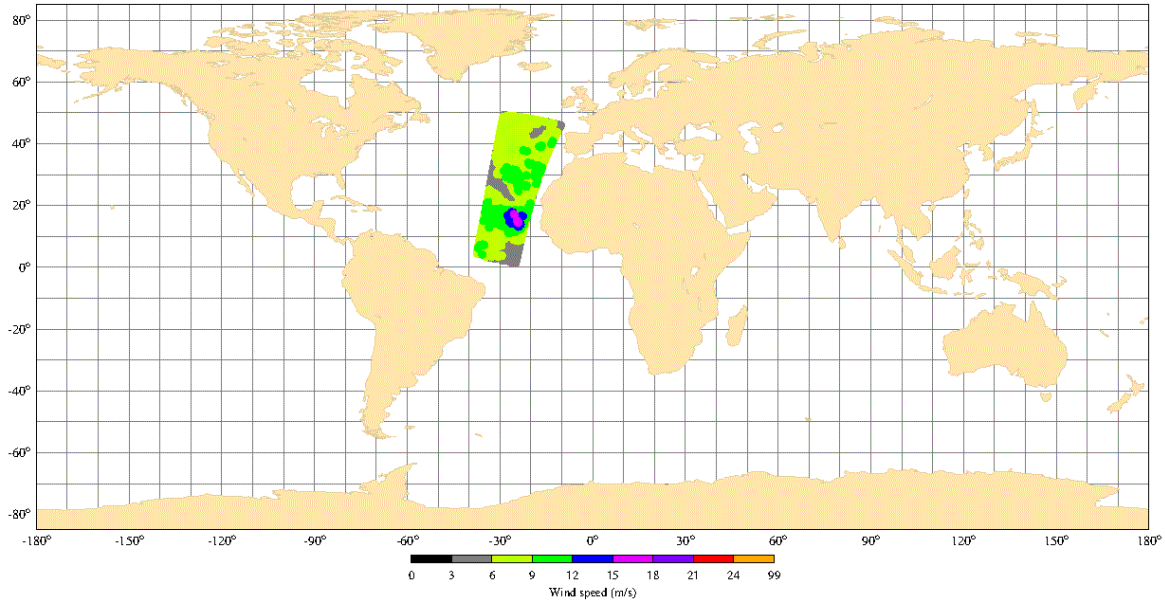
Figure 1.3 Global coverage of the test run. Wind speed results for the OSCAT 50 km product are shown.

Directory `genscat/support/bufr` contains a test program named `test_modules`. It is invoked by the genscat make system to construct the BUFR tables required by PenWP, but it can also be used to test the genscat BUFR module. The program is used as follows:

`test_modules [BUFRinput]`

where `BUFRinput` is the BUFR input file.

If omitted, the program uses as default input the file `testreading.bufr` in directory `genscat/support/bufr`. The output is written to a BUFR file named `testwriting.bufr`. The directory also contains a shell script named `run_test_modules` that sets the environment variables required and executes the program. Further information can be found in the comment lines of the source code of `test_modules`.

Directory `genscat/support/grib` contains test programs named `test_read_GRIB1`, `test_read_GRIB2` and `test_read_GRIB3`. The programs can be run from the command line and read in the GRIB file `testfile.grib` in directory `genscat/support/grib`. Some properties of this file are written to ASCII output files. Note that the environment variable `$GRIB_DEFINITION_PATH` needs to be set to directory `(…)/genscat/support/grib/definitions`.

Subdirectories `Compiler_Features`, `convert`, `ErrorHandler`, `singletonfft`, `file`, `BFGS`, `num`, `hdf5`, `sort` and `datetime` of `genscat/support` contain test programs for the module in that subdirectory. The test programs write their result to the standard output. In some cases, a copy of the output is contained in the `.output` files for comparison. Table 3.1 gives an overview of the genscat test programs.

Subdirectory	Program name	Output file	Remarks
bufr	test_modules	testwriting.bufr	Part of make system
grib	test_read_GRIB*	several	GRIB handling
Compiler_Features	TestCompiler_Features	-	Command line handling
convert	test_convert	test_convert.output	Wind speed conversion
ErrorHandler	TestErrorHandler	-	Error handling
singletonfft	TestSingleton	-	Fast Fourier Transform
file	TestLunManager	TestLunManager.output	File management
BFGS	Test_BFGS	-	Minimization
num	test_numerics	test_numerics.output	Numerical issues
hdf5	TestHDF5	testfile.h5	Read/write HDF5 files
sort	SortModTest	SortModTest.output	Array sorting
datetime	TestDateTimeMod	TestDateTimeMod.output	Date and time conversion

Table 3.1 Test programs in genscat/support.

3.5 Documentation

Directory `penwp/doc` contains documentation on PenWP, including this document. Further information can be found in the readme text files, and in the comments in scripts, Makefiles and source code.

References

[1]   Verhoef, A., Vogelzang, J., Verspeek, J. and Stoffelen, A., 2017,
PenWP Product Specification, Report NWPSAF-KN-DS-002, EUMETSAT.

[2]   Verhoef, A., Vogelzang, J., Verspeek, J. and Stoffelen, A., 2017,
PenWP Top Level Design, Report NWPSAF-KN-DS-001, EUMETSAT.

[3]   Freilich, M.H., 1999,
The SeaWinds Algorithm Theoretical Basis Document, NASA, (available on
http://eospso.gsfc.nasa.gov/sites/default/files/atbd/atbd-sws-01.pdf).

[4]   Padia, K, 2010,
Oceansat-2 Scatterometer algorithms for sigma-0, processing and products format,
Version 1.1, April 2010, ISRO.

[5]   Verhoef, A., Vogelzang, J., Verspeek, J. and Stoffelen, A., 2017,
PenWP Test Report, Report NWPSAF-KN-TV-008, EUMETSAT.

[6]   Stoffelen, A. and M. Portabella, 2006,
On Bayesian Scatterometer Wind Inversion, IEEE Transactions on Geoscience and
Remote Sensing, 44, 6, 1523-1533, doi:10.1109/TGRS.2005.862502.

[7]   Portabella, M. and Stoffelen, A., 2004,
A probabilistic approach for SeaWinds Data Assimilation, Quart. J. Royal Meteor. Soc.,
130, pp. 127-152.

[8]   Portabella, M. and Stoffelen, A., 2001,
Rain Detection and Quality Control of SeaWinds, Journal of Atm. Oceanic Technol., **18**,
pp. 1171-1183.

[9]   Portabella, M., 2002,
Wind field retrieval from satellite radar systems, PhD thesis, University of Barcelona.
(Available on http://www.knmi.nl/scatterometer/publications/).

[10]   Vogelzang, J., Stoffelen, A., Verhoef, A., de Vries, J. and Bonekamp, H., 2009,
*Validation of two-dimensional variational ambiguity removal on SeaWinds scatterometer
data*, J. Atm. Oceanic Technol., 7, 2009, 26, 1229-1245,
doi:10.1175/2008JTECHA1232.1.

[11]   Vogelzang, J., and Stoffelen, A., 2012,
NWP model error structure functions obtained from scatterometer winds, IEEE Trans.
Geosci. Rem. Sens., 50, 2012, 7, 2525-2533, doi: 10.1109/TGRS.2011.2168407.

[12]   Vogelzang, J. and Stoffelen, A., 2011,
Wind Bias Correction Guide, Report NWPSAF, EUMETSAT.

[13]   Vogelzang, J. and Stoffelen, A., 2015,
         High resolution data-assimilation guide, Report NWPSAF, EUMETSAT

Appendix A: Acronyms

Name	Description
ASCAT	Advanced SCATterometer on MetOp
AWDP	ASCAT Wind Data Processor
BUFR	Binary Universal Form for the Representation of data
C-band	Radar wavelength at about 5 cm
ERS	European Remote Sensing satellites
ECMWF	European Centre for Medium-range Weather Forecasts
EUMETSAT	European Organization for the Exploitation of Meteorological Satellites
genscat	generic scatterometer software routines
GMF	Geophysical model function
HDF5	Hierarchical Data Format version 5
ISS	International Space Station
KNMI	Koninklijk Nederlands Meteorologisch Instituut (Royal Netherlands Meteorological Institute)
Ku-band	Radar wavelength at about 2 cm
L1b	Level 1b product
LUT	Look up table
Metop	Meteorological operational Satellite
MLE	Maximum Likelihood Estimator
MSS	Multiple Solution Scheme
NCEP	United States National Centers for Environmental Prediction
NOAA	United States National Oceanic and Atmospheric Administration
NRCS	Normalized Radar Cross-Section (σ^0)
NWP	Numerical Weather Prediction
OSI	Ocean and Sea Ice
OWDP	OSCAT Wind Data Processor
PenWP	Pencil Beam Wind Processor
QC	Quality Control
SAF	Satellite Application Facility
SDP	SeaWinds Data Processor
SST	Sea Surface Temperature
WVC	Wind Vector Cell, also called node or cell

Table A.1 List of acronyms.

Appendix B: Development of PenWP

Software code for processing satellite data may become very complex. On the one hand, it consists of code related to the technical details of the satellite and instruments; on the other hand, the code drives complex algorithms to create the geophysical end products. Therefore, the EUMETSAT Satellite Application Facility (SAF) project for Numerical Weather Prediction (NWP) has included some explicit activities aiming at enhancing the modularity, readability and portability of the processing code.

For many years, the KNMI satellite observation research group has been developing processing code to supply Near Real Time (NRT) level 2 surface wind products based on the ERS, SeaWinds, ASCAT and OSCAT scatterometer level 1b Normalized Radar Cross Section data (σ^0). This work is coordinated and supervised by Ad Stoffelen. In the beginning only an adaptation of his ERS code existed. Later Marcos Portabella and Julia Figa added modifications and extensions to improve, e.g., the wind retrieval and quality control algorithms. In 2003 John de Vries finished the first official release of a processor within the NWP SAF. This processor was called the QuikSCAT Data Processor (QDP).

Meanwhile, Jos de Kloe has been updating the code for ERS scatterometer wind processing. For many parts of the process steps (e.g., the BUFR handling and part of the wind retrieval) a large overlap with SeaWinds Data processing coding exists. The KNMI Scatterometer Team is working towards generic NRT scatterometer processing. As a result, a new modular processing code for SeaWinds data was developed within the NWP SAF: the SeaWinds Data Processor (SDP) as successor of QDP. Based on the generic code already available for SeaWinds and ERS processing, a new ASCAT Wind Data Processor (AWDP) and later on an OSCAT Wind Data Processor (OWDP) were developed.

After SeaWinds on QuikSCAT and OSCAT on Oceansat-2 also HSCAT on HY-2A was launched, being the third Ku-band rotating pencil beam scatterometer in space. In 2014 RapidScat, built on the heritage of SeaWinds, was installed on the International Space Station. More instruments with similar configuration are planned for the (near) future. This has encouraged us to start the development of a generic wind processor being able to compute wind products for all these instruments: the Pencil beam Wind Processor (PenWP). The PenWP software is intended to replace both SDP and OWDP. It is already capable to process SeaWinds, OSCAT, HSCAT and RapidScat data and will be adapted when future instrument data become available.

Many persons contributed (directly or indirectly) to the development of the scatterometer software at KNMI: Hans Bonekamp, Jos de Kloe, Marcos Portabella, Ad Stoffelen, Anton Verhoef, Jeroen Verspeek, Jur Vogelzang and John de Vries are (in alphabetical order) the most important contributors.

Appendix C: HDF5 library copyright statement

This is the contents of the file called `COPYING` that is provided with the HDF Group software library and utilities. The text is also on http://www.hdfgroup.org/HDF5/doc/Copyright.html

```
Copyright Notice and License Terms for
HDF5 (Hierarchical Data Format 5) Software Library and Utilities
-----------------------------------------------------------------------------


HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright 2006-2009 by The HDF Group.

NCSA HDF5 (Hierarchical Data Format 5) Software Library and Utilities
Copyright 1998-2006 by the Board of Trustees of the University of Illinois.


All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted for any purpose (including commercial purposes)
provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice,
   this list of conditions, and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice,
   this list of conditions, and the following disclaimer in the documentation
   and/or materials provided with the distribution.

3. In addition, redistributions of modified forms of the source or binary
   code must carry prominent notices stating that the original code was
   changed and the date of the change.

4. All publications or advertising materials mentioning features or use of
   this software are asked, but not required, to acknowledge that it was
   developed by The HDF Group and by the National Center for Supercomputing
   Applications at the University of Illinois at Urbana-Champaign and
   credit the contributors.

5. Neither the name of The HDF Group, the name of the University, nor the
   name of any Contributor may be used to endorse or promote products derived
   from this software without specific prior written permission from
   The HDF Group, the University, or the Contributor, respectively.

DISCLAIMER:
THIS SOFTWARE IS PROVIDED BY THE HDF GROUP AND THE CONTRIBUTORS
"AS IS" WITH NO WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED.  In no
event shall The HDF Group or the Contributors be liable for any damages
suffered by the users arising out of the use of this software, even if
advised of the possibility of such damage.
```


Contributors: National Center for Supercomputing Applications (NCSA) at
the University of Illinois, Fortner Software, Unidata Program Center (netCDF),
The Independent JPEG Group (JPEG), Jean-loup Gailly and Mark Adler (gzip),
and Digital Equipment Corporation (DEC).

Portions of HDF5 were developed with support from the University of
California, Lawrence Livermore National Laboratory (UC LLNL).
The following statement applies to those portions of the product and must
be retained in any redistribution of source code, binaries, documentation,
and/or accompanying materials:

   This work was partially produced at the University of California,
   Lawrence Livermore National Laboratory (UC LLNL) under contract
   no. W-7405-ENG-48 (Contract 48) between the U.S. Department of Energy
   (DOE) and The Regents of the University of California (University)
   for the operation of UC LLNL.

   DISCLAIMER:
   This work was prepared as an account of work sponsored by an agency of
   the United States Government. Neither the United States Government nor
   the University of California nor any of their employees, makes any
   warranty, express or implied, or assumes any liability or responsibility
   for the accuracy, completeness, or usefulness of any information,
   apparatus, product, or process disclosed, or represents that its use
   would not infringe privately- owned rights. Reference herein to any
   specific commercial products, process, or service by trade name,
   trademark, manufacturer, or otherwise, does not necessarily constitute
   or imply its endorsement, recommendation, or favoring by the United
   States Government or the University of California. The views and
   opinions of authors expressed herein do not necessarily state or reflect
   those of the United States Government or the University of California,
   and shall not be used for advertising or product endorsement purposes.