RADS version 3.1

User Manual and Format Specification

Remko Scharroo

25 October 2012

This document was typeset with \LaTeX 2_ε.
The layout was designed by Remko Scharroo © 1993–2002

Contents

Chapter 1
Introduction

This document describes the layout, content, and use of the Radar Altimeter Database System (RADS), Version 3.0, that was developed by the Delft Institute for Earth-Oriented Space Research and the NOAA Laboratory for Satellite Altimetry. Apart from actual altimeter data, RADS provides a suite of applications and subroutines that simplify the reading, editing and handling of data from various radar altimeters. Although the actual content and layout of the underlying data products do not have to be identical for all altimeters, the user interface is. Also, the data base is easily expandable with additional data and/or additional corrections without impact to the user interface, or to the software in general. In fact, only in very few cases the software will need to be adjusted and recompiled, in even fewer cases adjustments to the actual tools will be required.

The data base consists of one meta file and one or more data files per satellite pass (half a revolution starting and ending close to the poles). Ascending passes have odd numbers, descending passes even numbers. The pass numbering increases consecutively within a repeat cycle. Which pass is number 1 is based on the longitude of the equator crossing (ascending node). Pass 1 has the smallest positive longitude of the ascending node.

The data files contain the actual (binary) data. The meta files describe the contents (data type, units, creation history, etc.) of the data in one (or possibly more) data files. There is only one meta file for each pass. The naming convention for the files is `SSpPPPPcCCC.nc`, where `SS` is an abbreviation for the satellite (altimeter), `PPPP` is the pass number, `CCC` is the cycle number, and `nc` is the extension (for netCDF data files).

The data files are grouped in one directory for each cycle, named `cCCC`. These cycle directories are then grouped into one directory for each mission phase, which are finally part of one directory per satellite. For example, the data file for pass 801 of cycle 150 in ERS-1's tandem mission is `$RADSDATAROOT/e1/g/c150/e1p0801c150.nc`, where `$RADSDATAROOT` is the root directory of the RADS data base. More details on the data base can be found in Appendix B.

To read and manipulate the data, you can use standard netCDF tools, like `ncdump` (that comes with the netCDF package) GMT (Generic Mapping Tools), nco (NetCDF Operators). But more suitable is the use of the RADS subroutine library and programs. The library is the basis for all data utilities provided with RADS and can also be used to create other programs to the user's convenience. For a description on how to create your own program see Chapter 5. The description of each of the subroutines is provided in Appendix A. In addition, a number of handy utilities are provided to do some of the most essential jobs Chapter 4.

Whether you are using the routines, or the provided utilities, you will have to know how the data handling system of RADS works. It is not essential that you understand the intrinsics of the data files, but it is highly recommended that you familiarise yourself with the way the data can be manipulated, selected and edited *on the fly* by the RADS routines. Basically, the RADS routines can take you a lot of work out of your hands, provided you have read Chapter 3.

Before going into the details of RADS, the software and the data have to be installed on your computer. Chapter 2 guides you through the installation process.

Chapter 2
Installation

In order to efficiently work with the RADS data base you are required to install the software (subroutine library, utilities, scripts and some supporting files). Also, at least part of the data base needs to be copied onto your hard disk (or another mounted device).

2.1 Prerequisites

In order to install and run the RADS software you need a few things installed on your system:

- A C compiler and a Fortran 77 (preferably Fortran 90) compiler.
- The netCDF library (version 3.6.2 or newer) and include files compiled with at least the C and Fortran 77 interface. Please figure out where to find the include files and the libraries before you continue.
- The rsync program.
- Optionally, the cvs program.

2.2 Software installation

The software can be downloaded from the RADS server using rsync or by using CVS. The two methods are described below. You can put the software anywhere you like. We will later configure where things will be installed. After downloading the software, continue with the configuration and compilation steps in Sections 2.2.3 and 2.2.4.

It is recommended to regularly check for updates of the RADS software and recompile if necessary.

2.2.1 Software download with rsync

Rsync is a mirroring tool for files across different machines. It does not do version control and will simply overwrite the current content.

You need to have at least the executable rsync installed on your system to use rsync. In case of Linux machines, simply install the rsync package available on most distributions. Rsync comes standard with Mac OS X, or can be obtained from `rsync.samba.org`.

Rsync will download the data from the rsync server at the Delft University of Technology in The Netherlands. This server is setup such that it will allow you to access only the RADS data and software. It will not allow you to log in to the server as a common user. Thus, setting up ssh key pairs is not possible.

Now you can copy the current version and previous versions of the software using the following commands:

```
rsync -avz --delete radsuser@rads.tudelft.nl::rads/code .
```

You will be asked to enter the password of radsuser before installation continues.

If you need to do this regularly and you do not want to enter the password every time, you can set up the environment variable `RSYNC_PASSWD` by one of the following methods (depending on the shell):

```
export RSYNC_PASSWD=radspasswd    # under sh or bash
setenv RSYNC_PASSWD radspasswd    # under csh or tcsh
```

If you are using the bash or sh shell, you can do it all in one line:

```
RSYNC_PASSWD=radspasswd \
   rsync -avz --delete radsuser@rads.tudelft.nl::rads/code .
```

Obviously, the password is **not** simply `radspasswd`.

The directory code will contain a number of tarballs, each containing the date on which they were created. There will also be a link to the latest version. For example:

```
lrwxrwxrwx 1 rads users     27 May  6 19:47 rads_source.tar.gz ->
                                             rads_source_20100506.tar.gz
-rw-r--r-- 1 rads users 459275 May  6 19:47 rads_source_20100506.tar.gz
```

You can extract the software in place, or anywhere you want by running

```
gzip -cd rads_source.tar.gz | tar -xvf-
```

This will create a directory called `src`.

2.2.2 Software mirroring with CVS

CVS is a version control system that helps to administrate software development projects on distributed systems (or at least by distributed users), avoiding problems of accidentally wiping out each others changes. Also, it is a very practical tool for distributing trees of software to others, who then can make their own changes without running the risk of accidentally overwriting them when a new update is provided. CVS can merge those changes, and alerts you of that happening.

You need to have at least the executable cvs installed on your system to use CVS. This program comes installed by default on Mac OS X and most Linux and Unix systems.

First you need to “login” to the CVS server. This needs to be done only once. Thereafter, the password gets stored (encrypted) in the file `.cvspass` in your home directory. To “login” use:

```
cvs -d:pserver:radsuser@rads.tudelft.nl:/home/rads/.cvs login
```

and type in the regular password for radsuser.

Now you can checkout a copy of the current version of the software using the following command:

```
cvs -d:pserver:radsuser@rads.tudelft.nl:/home/rads/.cvs co src
```

This will create a directory `src` whereever you were located at that time.

Later on you can bring the source on your machine up to date by going into the `src` directory and executing:

```
cvs update -Pd
```

2.2.3 Software configuration

Now we are going to determine where the software executables, library, and data is going to be stored. For this we run the configure in the `src` directory. This program also determines what compilers you have and what special options are needed for your platform.

By default, configure will install everything under the directory were you extracted the `src` directory. It will create directories:

bin for the executables (both binaries and scripts)

include for the include files to be used with all RADS programs

lib for the RADS library

share for the system independent data: the satellite data, namelists, and other configuration files. This one particularly, you might want to put somewhere else, on a dedicated disk, for example.

The configure program also helps you to set up the location of the netCDF software, that we need to know about before we can compile. Run, for example,

```
cd src
configure --with-netcdf-inc=/sw/lib/netcdf-gfortran/include \
    --with-netcdf-lib=/sw/lib:/sw/lib/netcdf-gfortran/lib \
    --prefix=/usr/local --datarootdir=/rads/data
```

The first option to configure identifies the directory where we can find `netcdf.inc`. The second option specifies the two directories that contain the netCDF C library (`libnetcdf`) and netCDF Fortran library (`libnetcdff`), separated by a colon. If these two are merged, or in one directory, you can just use one directory name. The third option specifies that the `bin`, `include` and `lib` directory are to be put under `/usr/local`. The last option specifies the directory for the data (which could be on a server for more systems to use).

The configure program also tests if your Fortran compiler is ready for Fortran 90 and can compile with the netCDF library. If you have problems, you may need to review the options you gave to configure, and make sure that configure picked the same compiler that was used to compile the netCDF library.

Run `configure --help` to get more info.

2.2.4 Software compilation

Now that your system is configured, it should be easy to just run make install in the `src` directory. It will first compile the programs, then install them in the right places. Note that it only installs the software, not the system independent data or the namelists.

If you have problems compiling, you might need to tweak the configuration files `src/include/config.mk`, `src/include/config.h`, or `src/include/config.inc`. Please let me know about it, so that I can change the configure program accordingly.

Running make install creates and installs executables of the software in the subdirectories `lib`, `utils`, `max2`, and `support`. These directories contain:

lib: library of the RADS subroutines;

utils: basic RADS utilities, explained in Chapter 4;

max2: crossover generation program, explained in Section 4.5;

support: supporting tools that manipulate the output of RADS utilities, and do not actually use the RADS library.

The remaining subdirectories contain:

tools: tools to create the data base. These programs require other routines, not provided with RADS, to compile, or use other data bases in order to run. They are provided to you to scan the source for what is done to the data. You should **not** attempt to compile and run the programs.

devel: utilities under development. They may or may not compile or run properly. Use at your own risk.

Now you can continue with the mirroring of the data files.

2.3 Mirroring the Data Base

RADS now nears 200 GBytes of data. It is impossible to copy all of it in one go, or copy all of it every time that updates have been made. To facilitate the updating, it is recommended to use the rsync program. This program will determine by it self which files are updated and will update only those. In fact, it will transfer only those parts of the files that are actually changed. This provides a significant speed benefit when, for example, an extra data field is added. See Section 2.2.1 for more information on rsync.

First you need to get the `nml` directory. This one contains information about defaults settings in parsing the data base. Assuming you have decided to put the RADS data in `/rads/data`, issue the following commands to retrieve this directory.

```
cd /rads/data
rsync -avz --delete radsuser@rads.tudelft.nl::rads/data/nml .
```

In the same way you can retrieve, for example, the Jason-2 data. Simply replace `nml` by `j2` in the above. See Table 3.1 for a list of all the 2-letter satellite abbreviations.

If you are patient, and want to get all of the data at once, you can perform the following commands:

```
cd /rads
rsync -avz --delete radsuser@rads.tudelft.nl::rads/data .
```

If, for whatever reason, the mirroring is interrupted, you can simply start it again, and it will continue where it left off. If you have a recent version of the rsync program, we recommend that you use the option `--del` instead of `--delete`, as it speeds up the process significantly.

2.4 Mirroring of additional information

There are two more directories that may be of interest, but are not essential. The `/rads/text` directory contains this manual, the `/rads/tables` directory contains a number of lists: lists of the time intervals of passes and cycles, and lists of the data available for each satellite.

To mirror both directories use rsync:

```
cd /rads
rsync -avz --delete \
    radsuser@rads.tudelft.nl::"rads/text rads/tables" .
```

Chapter 3

RADS Data Management

This Chapter describes the basic functionalities of the data management system of RADS. These functionalities are part of the RADS utilities as well as the RADS subroutine library on which the utilities are based.

3.1 Preparations before using the data base

The use of RADS starts with the definition of the environment variable `RADSDATAROOT`, such that it points to the root of the RADS data base. For example (for csh users):

```
setenv RADSDATAROOT /rads/data
```

If you already specified this directory in the configure step, then you do not have to set the `RADSDATAROOT` directory.

It is also practical to include `/usr/local/bin` (or where ever you installed the binaries) in your executable search path, so that existing executables can be used. However, this is not essential when you make your own software.

3.2 Common functionalities

A subroutine library is created to facilitate the data reading, conversion to SI units, editing and the construction of sea level anomalies. Based on these routines, several programs (utilities) are created to list or manipulate the contents of the data base. Since these programs share the same routines, much of their functionalities are the same, as well as their user interface. A second set of programs (tools) are available to create the data base. These programs will be of less concern to the users, and are not yet explained in this manual.

Common to all RADS utilities is the internal data selection, editing, and the ability to construct the sea level anomaly (and corrections fields) *on the fly*. Since the construction of the sea level anomaly is so important in many applications of altimetry, this functionality is provided within the RADS software library, so that a user, even when writing her/his own program, does not have to generate code to do so. At the same time that the sea level anomaly is constructed, the data is edited based on system-wide, user, or local preferences. These preferences can be specified at several levels, either in Fortran namelists, or as command line options. The order in which the preferences are processed is the following:

Altimeter	Abbreviation	Number	Meta file indication
GEOS 3	g3	1	GEOS-3
Seasat	sa	2	SEASAT
Geosat	gs	3	GEOSAT
ERS-1	e1	4	ERS-1
TOPEX	tx	5	TOPEX
Poseidon	pn	6	POSEIDON
ERS-2	e2	7	ERS-2
GFO	g1	8	GFO-1
Jason-1	j1	9	JASON-1
Envisat	n1	10	ENVISAT1
Jason-2	j2	11	JASON-2
CryoSat2	c2	12	CRYOSAT2

Table 3.1 Abbreviation and meta file indication used for the various altimeters.

- System-wide general preferences, found in `$RADSDATAROOT/nml/getraw.nml`.
- Satellite-specific and mission-specific system-wide preferences, found in the files `getraw_SS.nml` and `getraw_SSF.nml` in the directory `$RADSDATAROOT/nml`, where `SS` is the 2-letter abbreviation for the satellite, as specified in Table 3.1, and `F` is a 1-letter indication of the mission phase.
- General user preferences, found in a file `getraw.nml` in the directory `~/.rads`.
- Satellite-specific and mission-specific user preferences, found in the files `getraw_SS.nml` and `getraw_SSF.nml` in the directory `~/.rads`.
- General local preferences, found in a file `getraw.nml` in the current working directory.
- Satellite-specific and mission-specific local preferences, found in the files `getraw_SS.nml` and `getraw_SSF.nml` in the current working directory.
- Command line arguments.

So the command line arguments overrule the local preferences and they overrule the user preferences and eventually the system-wide preferences.

3.3 Data fields and defaults

Before continuing to explain how the user preferences can be coded, it is important to know which *data fields* are available for which satellites, how they are edited, and how the sea level anomaly is created. Each data field is specified by a unique *data field descriptor*, a number between 0 and 9999. Some of there numbers link directly to a data field (for example, the latitude of the measurement), others to an algorithm that computes the data field from others (like the time of day). A special case is the sea level anomaly.

First, we have to distinguish between the numbers 0, 1–99, and 100–9999. Each of these ranges have special meaning. The higher range of numbers link to specific data fields, like ”1207” is the GOT00.2 ocean tide. This number can be

data	selection		default values				
type	code	name [unit]	option	factor	min	max	sats
0		sea level anomaly [m]			-5	+5	gs e1 e2 tx pn g1 j1
					-4.5	+5.5	n1
1		time [s]	1	0	-1d20	+1d20	
	101	*relative to 1985.0*					
	102	*relative to equator crossing*					
	103	*local solar time of day*					
	104	*in format [yyyymmddhhmmss.ff]*					
	105	*as [MJD]*					
2	201	latitude [deg]	1	0	-90	+90	
3	301	longitude [deg]	1	0	-180	+180	
4		orbital altitude (TOPEX ellipsoid) [m]	11	1	1300d3	1400d3	tx pn
	401	*JGM-3 gravity*	2	1	750d3	850d3	e1 e2
	402	*DGM-E04 gravity*	12	1	1300d3	1400d3	j1
	403	*GFZ orbit*	10	1	750d3	850d3	g1
	404	*CNES orbit*	12	1	750d3	850d3	n1
	405	*PGS7727 gravity*	11	1	750d3	850d3	gs
	406	*GRIM5-C1 gravity*					
	409	*EIGEN GRACE01S gravity*					
	410	*PGS7777 gravity*					
	411	*GGM02C gravity (ITRF2000)*					
	412	*EIGEN CG03C gravity*					
	413	*GGM02C gravity (ITRF2005)*					
	414	*EIGEN GL04C gravity (GDR C′)*					
	415	*EIGEN GL04C gravity (GSFC)*					
	416	*GPS Near Real Time orbit (JPL)*					
5	501	orbital altitude rate [m/s]	-1	0	-30	+30	e1 e2 g1 n1 j1
	502	range rate [m/s]	0				tx pn gs
6	601	altimeter range corrected	1	-1	1300d3	1400d3	tx pn j1
		for instrument effects [m]	1	-1	750d3	850d3	e1 e2 gs g1 n1
7		dry tropo correction [m]	1	-1	-2.4	-2.1	gs e1 e2 tx pn j1 n1
	701	*ECMWF model*	2	-1	-2.4	-2.1	g1
	702	*NCEP/NCAR model*					
	708	*NCEP reanalysis 2*					
	709	*ECMWF ERA-interim*					
8		wet tropo correction [m]	1	-1	-0.6	0.0	e1 e2 tx pn g1 j1
	801	*radiometer measurement*	3	-1	-0.6	0.0	gs
	802	*ECMWF model*	1	-1	-0.6	0.05	n1
	803	*NCEP/NCAR model*					
	804	*NASA NVAP model*					
	805	*TOVS/SSMI data*					
	807	*TOVS/NCEP hybrid*					
	808	*NCEP reanalysis 2*					
	809	*ECMWF ERA-interim*					
9		ionospheric correction [m]	8	-1	-0.40	+0.04	gs e1 e2 pn
	901	*dual-frequency altimeter*	3	-1	-0.40	+0.04	tx j1 n1
	902	*Bent model*	6	-1	-0.40	+0.04	g1
	903	*smoothed dual-freq value*					
	904	*DORIS model*					
	905	*CODE GIM model*					
	906	*JPL GIM model*					
	907	*IRI2007 model*					
	908	*NIC08 model*					
10		inverse baro correction [m]	4	-1	-1	+1	
	1001	*local pressure - 1013.3 mbar*					
	1002	*local - global mean pressure*					
	1003	*1013.3 mbar - global mean pressure*					
	1004	*MOG2D model*					
	1005	*MOG2D mean (1993-2007)*					
11	1101	solid earth tide [m]	1	-1	-1	+1	

Table 3.2 Data fields, flavours, and default settings.

data	selection		default values				
type	code	name [unit]	option	factor	min	max	sats
12		ocean tide [m]	13	-1	-5	+5	
	1201	*CSR 4.0 model*					
	1202	*CSR 4.0 model (incl loading)*					
	1207	*GOT00.2 model*					
	1208	*GOT00.2 model (incl loading)*					
	1211	*FES2002 model*					
	1212	*FES2002 model (incl loading)*					
	1213	*FES2004 model*					
	1214	*FES2004 model (incl loading)*					
	1215	*WebTide model (local)*					
	1217	*GOT4.7 model*					
	1299	*long-periodic equil. tide*					
13		load tide [m]	13	-1	-0.5	+0.5	
	1301	*CSR 4.0 model*					
	1307	*GOT00.2 model*					
	1311	*FES2002 model*					
	1313	*FES2004 model*					
	1317	*GOT4.7 model*					
14	1401	pole tide [m]	1	-1	-0.1	+0.1	
15		sea state bias [m]	1	-1	-1	+1	
	1501	*BM3/BM4 model*					all
	1502	*CLS non-parametric model*					tx j1 n1
	1503	*CSR-A/B model*					
	1504	*hybrid model*					pn n1 gs g1
16		geoid or mss height [m]	5	-1	-200	+200	
	1601	*EGM96 geoid height*					
	1602	*OSU MSS95 height*					
	1603	*GSFC00.1 MSS height*					
	1604	*T/P along-track MSS height*					
	1605	*CLS01 MSS height*					
	1608	*GGM02C geoid*					
	1610	*EGM2008 geoid*					
	1611	*EGM2008 mean sea surface*					
	1612	*DNSC08 mean sea surface*					
17		significant wave height [m]	1	0	0	8	
	1701	*Ku-band value*					
	1702	*C/S-band value*					
18		backscatter coefficient [dB]	1	0	6	27	
	1801	*Ku-band value*					
	1802	*C/S-band value*					
19		wind speed [m/s]	1	0	0	30	
	1901	*altimeter estimate*					
	1902	*radiometer estimate*					
	1903	*model U-component*					
	1904	*model V-component*					
20		sigma range [m]	1	0	0	0.05	tx
	2001	*1-Hz error estimate*	1	0	0	0.09	e1 e2 pn gs g1 n1
	2002	*10/20-Hz standard deviation*	1	0	0	0.04	j1
21	2101	number of averaged measurements [-]	1	0	8.5	10.5	tx gs g1
			1	0	16.5	20.5	e1 e2 pn n1
			1	0	15.5	20.5	j1
22		ocean depth and land elevation [m]	-1	0	-1d20	+1d20	
	2201	*ETOPO2*					
	2202	*DTM2000.1*					
	2203	*DNSC08*					
	2204	*SRTM_30+*					

Table 3.2 Data fields, flavours, and default settings (cont'd).

data	selection		default values				
type	code	name [unit]	option	factor	min	max	sats
23		brightness temparatures	-2	0	-1d20	+1d20	e1 e2 tx pn g1 j1
	2301	... at 18 GHz [dB]					
	2302	... at 21-24 GHz [dB]					
	2303	... at 34-37 GHz [dB]					
24		peakiness [-]	1		1.5	1.8	n1
	2401	*Ku-band value*					
	2402	*C/S-band value*					
25		bit flags based on field 26					
	25nn	*bit nn of field 26*					
26	2601	engineering flags	1	0			Table 3.3
27		range biases (*) [m]	-1	0	-1d20	+1d20	e1 e2 gs n1
	2701	*constant bias*	0				tx pn g1
	2702	*SPTR bias / Internal cal (corr.)*					
	2703	*USO bias*					
	2704	*Doppler correction*					
	2705	*Ku-band SWH/attitude correction*					tx gs
	2706	*C-band SWH/attitude correction*					tx
	2707	*centre-of-mass correction*					tx
	2708	*datation bias correction*					j1
28		sigma SWH [m]	2	0	0	0.9	tx
	2801	*1-Hz error estimate*	2	0	0	2.1	e1 e2 g1 n1 j1
	2802	*10/20-Hz standard deviation*	0				pn gs
29		sigma sigma0 [dB]	-2	0	-1d20	+1d20	e1 e2 tx j1 n1 pn
	2901	*1-Hz error estimate*	0				g1 gs
	2902	*10/20-Hz standard deviation*					
30		off-nadir orientation angle	-2	0	-1d20	+1d20	
	3001	*from waveform [deg]*	-2	0	-1d20	+1d20	
	3002	*from waveform [deg^2]*	-3	0	-1d20	+1d20	all, except e1
	3003	*from platform [deg]*	-4	0	-1d20	+1d20	
	3004	*from platform [deg^2]*	2	0	-0.05	0.05	
31		significant wave height bias (*) [m]	-1	0	-1d20	+1d20	
	3101	*Ku-band bias*					tx pn
	3102	*C/S-band bias*					tx
	3103	*Ku-band net instr. correction*					tx
	3104	*C-band net instr. correction*					tx
32		backscatter coefficient bias (*) [dB]	-3	0	-1d20	+1d20	tx pn n1
	3201	*Ku-band bias*					
	3202	*C/S-band bias*					
	3203	*atmospheric atten. Ku-band*					
	3204	*atmospheric atten. C/S-band*					
33	3301	Liquid water content [kg/m^2]	-1	0	-1d20	+1d20	n1
34		House keeping / miscellaneous	-1	0	-1d20	+1d20	
	3401	*Frame counter [-]*					
	3402	*Mode word [-]*					
	3403	*Flag word [-]*					
	3404	*Receiver temperature [K]*					
	3405	*Vatt-T [V]*					
	3407	*Ku-band gate index [-]*					tx
	3408	*C-band gate index [-]*					tx
	3409	*Kappa [-]*					
35	3501	Rain/ice index [-]			0	2	n1
36	3601	Basin code [-]			0.5	255.5	all
37		Waveform data					
38	3801	Reference frame offset	1		-1	+1	all
39		Long period tides	-1	0	-1	+1	all
	3901	Long period equilibrium tide					all
	3902	Long period non-equilibrium tide					all

(*) Biases are already applied to the data values.

Table 3.2 Data fields, flavours, and default settings (cont'd).

data	selection		default values				
type	code	name [unit]	option	factor	min	max	sats
40	4001	Wave height variance [m^2]			-1d20	+1d20	
41	4101	First order moment of wave height [m^2/s]			-1d20	+1d20	
42	4201	Wave velocity variance [m^2/s^2]			-1d20	+1d20	
43	4301	Wave slope variance [$radian^2$]			-1d20	+1d20	
44	4401	Swell height [m]			-1d20	+1d20	
45	4501	Distance from coast [km]			-1d4	+1d4	all
46	4601	Sea ice classification [-]			-0.5	15.5	n1
47	4701	Sea ice concentation [%]			-0.5	100.5	all
48	4801	Sea surface temperatude [°C]			-2	+40	all
49	4901	GIA correction [m]			-1d20	+1d20	
99		track parameters			-1d20	+1d20	
	9901	*pass start time [SEC85]*					
	9902	*pass end time [SEC85]*					
	9903	*time of equator crossing [SEC85]*					
	9904	*longitude of equator crossing [deg]*					
	9905	*cycle number*					
	9906	*pass number*					
	9907	*number of measurements*					
	9999	*zero constant*					

(*) Biases are already applied to the data values.

Table 3.2 Data fields, flavours, and default settings (cont'd).

split into the *data type* "12", indicating "ocean tide", and its *flavour* "07", indicating "GOT00.2". A full list of the data field descriptors is given in Table 3.2.

The range of numbers 1–99 refers to the data type. So the number "12" refers to "ocean tide". Another number, called the *option* will then determine which flavour will be used. For example, if field 12 is given option 7, the GOT00.2 ocean tide (data field 1207) will be used for ocean tide. By referring to the data fields using the numbers 1–99, you do not have to specify which of the alternative data fields need to be used. This choice is already made in the default settings. For example, when you choose field 9, it may return the "smoothed dual-frequency ionosphere correction" (data field 903) for TOPEX, but the "Bent ionosphere correction" (data field 902) for Poseidon.

Finally, "0" is the sea level anomaly. It will be constructed as a linear combination of the various data fields available in a data file, partly based on the options set for the various correction fields.

Not all fields are available for all altimeter missions. Refer to Table 3.4 to see which fields are provided for which missions.

Table 3.2 also indicates the options (*i.e.*, the default flavours) for each of the data fields. These options may be different for the different satellites. Take, for example, the ionospheric correction (data type 9): for the dual-frequency altimeters (TOPEX, Envisat, and Jason-1), the smoothed dual-frequency value (`options(9)=3`) will be used by default; the Bent model (`options(9)=2`) is used for ERS-1, ERS-2 and Poseidon; the JPL GIM model is used for GFO (`options(9)=6`); the IRI95 model is used for Geosat.

The *factor* is a multiplication factor for the correction in the creation of the sea level anomaly. When its factor is +1 a data field will add to the sea surface height, when -1 it will be subtracted. So for the orbital altitude the factor is +1, thus `factors(4)=+1`. However, for example, the ionosphere correction is to be subtracted from the orbital altitude, hence the factor is -1 (`factors(9)=-1`).

A factor "0" will mean that a correction is neither added nor subtracted from

Bit	Description	sats
0	Hardware status	
	0=ok, 1=bad (fatal)	tx j1
	0=Side A, 1=Side B	n1
	0=ok, 1=bad (questionable)	pn
	0=LRM, 1=SAR (PLRM)	c2
1	Satellite on track: *0=ok, 1=out of dead band*	e1
	Attitude status: *0=nominal, 1=suspect*	e2 gs tx pn g1 j1
2	dH status: *0=nominal, 1=suspect*	gs
	Continental ice: *0=no, 1=yes*	j1 n1
	TMR Channel 21: *0=A, 1=B*	tx pn
3	Quality of dual-frequency ionosphere: *0=ok, 1=bad*	**tx n1**
4	Altimeter land flag based on 2′×2′ mask	
	0=water (ocean, enclosed seas, lakes), 1=land	**all**
5	Altimeter ocean/non-ocean flag based on 2′×2′ mask	
	0=ocean, 1=non-ocean (land, enclosed seas, lakes)	all
6	Radiometer land flag	
	1=land, 0=water	**e1 e2 tx pn g1 j1 n1**
7	Corruption of altimeter measurement	
	0=ok, 1=ice	**tx pn j1 g1**
	0=ok, 1=rain or ice	**n1**
8	Corruption of radiometer measurement	
	0=ok, 1=rain or ice	**e1 e2 tx pn**
	0=ok, 1=rain	**j1**
9	Quality of the 23.8/22/18.7 GHz brightness temperature: *0=ok, 1=bad*	**e1 e2 g1 j1 n1**
	Quality of TMR brightness temps, bit 0 of TMR_Bad: *0=ok, 1=bad*	**tx pn**
10	Quality of the 36.5/37/34.0 GHz brightness temperature: *0=ok, 1=bad*	**e1 e2 g1 j1 n1**
	Quality of TMR birghtness temps, bit 1 of TMR_Bad: *0=ok, 1=bad*	**tx pn**
11	Quality of the range estimate:	
	0=ok, 1=suspect	**e1 e2** tx pn **g1 j1** n1
	0=ok, 1=some 10-Hz invalid	gs
12	Quality of the wave height estimate: *0=ok, 1=suspect*	**e1 e2** tx pn **g1 j1**
	Sea state bias: *0=valid, 1=suspect*	gs
13	Quality of the backscatter coefficient estimate: *0=ok, 1=suspect*	**e1 e2** tx pn **g1 j1** n1
	Quality of windspeed measurement: *0=ok, 1=suspect*	gs
14	Tracking mode	
	0=normal, 1=preset	**e1 e2**
	0=normal, 1=coarse/acquisition	**gs tx pn j1**
	0=normal, 1=modelled USO correction	n1
15	Quality of the orbit	
	0=ok, 1=degraded	**e1 e2 pn j1 n1**

Table 3.3 Description of flags. Bold face indicates which flags are checked by default for each satellite.

the sea surface height, but may be considered as an edit criterion in the construction of the sea level anomaly. The significant wave height and backscatter coefficients are examples of this; they are very useful indicators of the data quality, but should not be added or subtracted from the sea level anomaly. Hence, `factors(17)=0, factors(18)=0`.

The columns "min" and "max" indicate the validity *limits* for this correction. For example, any ionospheric corrections beyond the range from -40 to +4 cm will be rejected, and set to NaN (Not-a-Number). This will also set the sea level anomaly to NaN. By the way, a positive ionosphere correction up to +4 cm is allowed because of the noise in the dual-frequency values.

In some cases a negative option is indicated, as in the case of the "sigma sigma-0" (data type 29). The "-1" reported here means that this data field is *not* used for the creation of the sea level anomaly, *nor* for its editing. However, if a

selection	Satellites							
code	gs	e1	tx	pn	e2	g1	j1	n1
101-105	●	●	●	●	●	●	●	●
201	●	●	●	●	●	●	●	●
301	●	●	●	●	●	●	●	●
401	●		●	●				
402		●			●			
404			●	●				●
409					a			
410						●		
411	●		●	●			●	
412							●	●
413			●	●			●	
501		●			●	●	●	●
601	●	●	●	●	●	●	●	●
701	●	●	●	●	●		●	●
702	●	●				●		
801		●	●	●	●	●	●	●
802		●	●	●	●		●	●
803	●	●				●		
804-805	●							
901			●				●	●
902		●	●	●	●			●
903			●				●	●
904			●	●			●	●
906			b	b	b	●	●	●
907-908	●	●	●	●	●	●	●	●
1001-1003	●	●	●	●	●	●	●	●
1004		c	●	●	●	●	●	●
1005	●			●	●	●	●	
1101	●	●	●	●	●	●	●	●
1207,1307		●			●			●
1213,1313	●	●	●	●	●	●	●	●
1215,1315	d	d	d	d	d	d	d	d
1217,1317	●		●	●		●	●	
1401	●	●	●	●	●	●	●	●
1501	●	●	●	●	●	●	●	●
1504	●		●	●		●	●	●
1601,1603		●			●			●
1605,1608	●	●	●	●	●	●	●	●
1610-1612	●		●	●		●	●	
1701	●	●	●	●	●	●	●	●
1702			●				●	●

selection	Satellites							
code	gs	e1	tx	pn	e2	g1	j1	n1
1801	●	●	●	●	●	●	●	●
1802			●				●	●
1901	●	●	●	●	●	●	●	●
1902							●	
1903-1904							●	●
2001-2002	●	●	●	●	●	●	●	●
2101	●	●	●	●	●	●	●	●
2201		●			●			●
2202								●
2203	●		●	●		●	●	
2301			●	●			●	
2302-2303		e	●	●	●	●	●	●
2401-2402								●
25xx	●	●	●	●	●	●	●	●
2601	●	●	●	●	●	●	●	●
2701	●	●			●			●
2702-2703	●	●			●			
2704								●
2801-2802		●	●		●	●	●	●
2901-2902		●	●	●	●		●	●
3001-3002	●		●	●	●	●	●	●
3003-3004				●			●	●
3101-3102		●						
3201-3202		●						
3203		●	●				●	●
3204								●
3601	●	●	●	●	●	●	●	●
3801	●	●	●	●	●	●	●	●
3901-3902	●		●	●		●	●	
4001-4401			f	f		f	f	

Remark	
●	Available throughout the mission
a	Cycles 70 and further
b	From September 1998 onward
c	From 1992 onward
d	Locally
e	Phases C and G only
f	Years 2000 through 2005 only

Table 3.4 Availability of data fields across the altimeter missions. Bullets (•) indicate availability throughout the mission, letters (a-f) refer to the remarks at the bottom of the table.

data field ”29” is requested, data field ”2901” (the 1-Hz standard deviation of the backscatter coefficient) will be returned. Since fields with a negative option are not used in the construction of the sea level anomaly, the factor is 0.

Finally, when the option is set to ”0”, the particular data field is not available. For example, the RADS files for TOPEX, Poseidon and Geosat do not contain the orbital altitude rate, so `factors(5)=0`.

For the construction of the sea level anomaly (data field descriptor 0) all fields with a positive option number are checked against the specified limits. When any field fails this check, no sea level anomaly is created. If a field passes the check, it is multiplied with the specified factor and added to the total sum. In the end, the created sea level anomaly is checked against its specified limits (default: -5 to 5 metres).

Another, less important, controllable feature is the output format for each data type. An attempt is made to predefine output formats that are most suitable for a data type. For example `formts(19)="f7.3"` tells that the wind speed will be printed with the `f7.3` format, meaning 7 characters including a decimal point and three decimal digits, allowing outputs like `29.999`. You can change this setting as explained in Section 3.5.

3.4 Special data elements

All data elements starting with 99 are special constants per pass, like pass and cycle number. See Table 3.2.

3.5 Namelists

The default settings for data ranges, options and multiplication factors are not hard coded, but controlled by *namelists* that are parsed by the routine `GETRAW_NML` which is called by the routine `GETRAW_INIT`, the essential first call to the RADS routines in any of the RADS utilities. The first namelist to be parsed is `$RADSDATAROOT/getraw.nml`, containing the system-wide satellite-independent settings. In this file you can find the default values for the arrays `options`, `factors`, and `limits`. For example, `limits(1,9)=-0.40` sets the lower limit for the editing of the ionosphere correction to -40 cm, `limits(2,9)=+0.04` set the upper limit to +4 cm.

`GETRAW_INIT` first loads, in sequence, the namelists `getraw.nml`, `getraw_SS.nml` and `getraw_SSF.nml` from the directory `$RADSDATAROOT/nml`, where `SS` is the 2-letter satellite abbreviation and `F` the 1-letter mission phase indicator. These three files contain the system-wide general, satellite- and mission phase-specific settings. The first namelist contains the settings common to all satellites, the second namelists contains settings for a particular satellite that are different from the common (or general) settings. The third file may contain certain settings for a single mission phase, as far as they differ from the general or satellite-specific settings.

Then, the user can place files `getraw.nml`, `getraw_SS.nml`, and/or `getraw_SSF.nml` in the directory `~/.rads`. These user-defined namelists subsequently overrule the previous system-wide settings settings and will be used any time the user runs a RADS program.

Finally, the user can create her/his local settings by creating files `getraw.nml`, `getraw_SS.nml`, and/or `getraw_SSF.nml` in the directory in which the program is executed. These settings will, again, overrule settings in the previous files and are only used when a program is run in that particular directory.

It is clear that these namelist files can be pretty small. For example:

```
 &getraw_nml
 limits(1,2)=28, limits(2,2)=48,
 limits(1,3)=-8, limits(2,3)=42,
 options(12)=7,
 options(13)=7,
 options(15)=-1,
 factors(15)=0,
 formts(19)="f6.3",
 &end
```

This will select altimeter data over the Mediterranean Sea, using the GOT00.2 ocean and loading tide instead of the default FES2004 model. Also, the sea state bias is not checked or included as correction. In principle, the addition `factors(15)=0` is superfluous, since `options(15)` is negative. The line `formts(19)="f5.3"` specifies the output format for wind speed, which is, actually, the default. Repeating the obvious, however, does not do any harm in RADS. Note the required space at the beginning of each line!

3.6 Functions

Functions were previous handled by FUNCT records. The use of FUNCT is now depreciated. This functionality will soon be removed from RADS.

Some data fields are defined as a function of one or two other fields. The most obvious function is making a linear combination of 2 other fields. These *functional* data fields are defined in a small system-wide RADS meta file `$RADSDATAROOT/nml/getraw.rmf`. An excerpt of this file is given below:

```
@RADS_RMF_V2.2
MATH  1208 '$1207 $1307 ADD =' 'ocean+load tide, GOT00.2 [m]'
MATH  2001 '$2002 $2101 2 SUB SQRT DIV =' 'norm std dev of range [m]'
```

This file specifies that the GOT00.2 combined ocean+load tide (data field 1208) is created by adding the GOT00.2 load tide (data field 1307) to the ocean tide (data field 1207). The normalised standard deviation of 1-Hz range (data field 2001) is created out of the 10-Hz or 20-Hz value (field 2002) and the number of elementary measurements (field 2101) as $\sigma/\sqrt{n-2}$. More information about how to craft `MATH` records is provided in Chapter C.

Similar to the namelists, RADS supports general, satellite- and mission-specific meta files (`getraw.rmf`, `getraw_SS.rmf` and `getraw_SSF.rmf`). These files are search in that order first in the directory `$RADSDATAROOT/nml` for system-wide settings, then in `~/.rads` for user-defined settings, then in the working directory for local settings.

Note that the user can also create her/his own data fields that are a function of other data fields. For example, to create a field (let's call it "1250") for the difference between the FES2004 and GOT00.2 ocean tide, the user can create a file `getraw.rmf` with the following content:

```
@RADS_RMF_V2.2
MATH 1250 '$1213 $1207 SUB =' 'ocean tide difference, FES2004-GOT00.2 [m]'
```

3.7 Time, latitude and longitude

To select a certain period, or region, limits on time, latitude and longitude can be specified. Based on these limits, the RADS library will skip passes that do not cross this area or do not have data in the specified period. Instead of reading all the data points, the software will use the meta data in the netCDF files to decide whether the pass can possibly cross the specified area and can possibly contain data within the given time period.

3.8 Flags

Field number 26 is an integer 16-bit word. These bits are based on the original GDR products and/or on external bits maps. An attempt is made to harmonise the meaning of each of the bits throughout the different altimeters, however some difference remain. See Table 3.3 for a complete description of the data flags and consult Table 3.2 for the bits that are tested by default.

The meaning of the limits for the flag field is different from the other limits. In stead of indicating the minimum and maximum value, `limits(1,26)` is an integer 16-bit word in which the bits are set that *should not* be set in the data field, while `limits(2,26)` is an integer 16-bit word in which the bits are set that *should* be set in the data field. For example, if `limits(1,26)=17` and `limits(2,26)=2`, data will be rejected in which bit 0 is set, bit 4 is set, or bit 1 is not set.

Field 25 is a special field. It is a function of the bit field 26. Field 25nn will be 1 if bit nn of field 26 is set, otherwise it is 0. Do not use field 25 for data editing purposes; the testing of the flag bits can be controlled by setting the limits of field 26, as explained in Section 3.5.

Chapter 4 Utilities

The utilities described in this Chapter are available in the directory `bin`. The source code can be found in `src/utils` and `src/max2`.

4.1 Common command line arguments

The RADS utilities share a number of command line arguments for selecting the data set and to indicate how the data set is to be sliced and manipulated. These arguments, of which only the **`sat=`** argument is required (the rest are optional) are explained in this Section.

When a program adds a different meaning to one of these arguments, it is explained in the program description in the next Sections. The order in which the arguments appear on the command line is generally irrelevant, and some may appear more than once. However, later appearances of the same argument may overrule earlier ones.

In the following, the different arguments are listed. When the argument has an optional part, it is shown within square brackets. The variable part of the arguments is written in *italics*.

4.1.1 Data set specifiers

`sat=`*`altimeter`*`[/`*`phase`*`]` or **`sat=`*`altimeter`*`[:`*`phase`*`]`** Specifies the altimeter mission and (optionally) its mission phase. The ***`altimeter`*** can be indicated by its two-character code listed in Table 3.1, or by its name. For example, **`e1`** and **`ERS-1`** are both allowed. The ***`phase`*** is a one-character indicator, which defaults to **`b`** for Geosat, **`g`** for ERS-1, **`b`** for Envisat, and **`a`** for all other satellites. For example:

`sat=ERS-1/c`	Selects ERS-1 phase C (Multi-disciplinary Phase)
`sat=tx`	Selects TOPEX (not Poseidon) data
`sat=e1`	Selects ERS-1 Tandem Mission data (Phase G)
`sat=gs/a`	Selects Geosat Geodetic Phase data

Note that both a slash (/) or a colon (:) can be used as a delimeter between the altimeter and the mission phase. This is the only argument that should always appear on the command line.

`cycle=`*`first`*`[,`*`last`*`]` Selects one or a range of cycle numbers. When only ***`first`*** is specified, one cycle is selected. For example:

`cycle=1`	Selects cycle 1
`cycle=1,6`	Selects cycles 1 through 6

When the **`cycle=`** argument is omitted, all of the cycles within the specified mission will be used.

`pass=`*`first`*`[,`*`last`*`[,`*`step`*`]]` Selects one or a range pass numbers. When only ***`first`*** is specified, one pass (per cycle) is selected. The ***`step`*** option can be added to skip passes. For example:

`pass=1`	Selects only the first pass of all selected cycles
`pass=1,1002`	Selects passes 1 through 1002 of all selected cycles
`pass=1,1002,2`	Selects only the ascending passes
`pass=2,1002,2`	Selects only the descending passes

When the **`pass=`** argument is omitted, all of the passes within the specified cycles will be used.

4.1.2 Data selection arguments

`nml=`*`namelist`* Loads ***`namelist`*** in additional to system-wide and user-defined namelists.

`ymd=`*`starttime`*`,`*`endtime`* Indicates the period of selection, where ***`starttime`*** and ***`endtime`*** are in the form [YY]YYMMDD[HHMMSS.SSS]. Other valid time selection arguments start with **`yod=`** for year/day-of-year in the form [YY]YYDDD.DDD, **`mjd=`** for Modified Julian Date, **`sec=`** for UTC seconds since 1.0 January 1985, and **`t=`** for an educated guess between these formats. The following examples all select data between 1 January 1999, 12:00 UTC and 31 December 2001, 00:00 UTC:

`ymd=990101.5,20011231`	Note that the century can be omitted
`doy=99001.5,2001365`	when using YMD or DOY.
`mjd=51179.5,52274.0`	With MJD or DOY, use only fractions of day,
`ymd=990101120000,011231`	but with YMD use also "HHMMSS".
`sec=441806400,536371200`	The old-fashioned seconds since 1985.

When omitted, the range specified by `limits(1,1)` and `limits(2,1)` in the namelists is used. By default, this is from the Big Bang to the Big Crunch.

`lat=`*`south`*`,`*`north`* Selects the latitude range (in degrees) for the data selection. The south to north range has to be specified on the interval [-90,90]. Examples:

`lat=-40,-20`	Select data between 40°S and 20°S
`lat=0,90`	Selects entire northern hemisphere

When omitted, the default range specified in the namelists by `limits(1,2)` and `limits(2,2)` is used. By default, this is -90 to +90.

`lon=`*`west`*`,`*`east`* Selects the longitude range (in degrees) for the data selection. The west to east range can be specified on the interval [0,360] or [-180,180] (or even [-360,0]), as long as the western limit has a longitude smaller than the eastern limit. Note that the output longitudes are influenced by the selection of either interval. Examples:

`lon=-40,-20`	Select data between 40°W and 20°W
`lon=320,340`	Same, but output will have only positive longitudes

When omitted, the default range specified in the namelists by `limits(1,3)` and `limits(2,3)` is used. By default, this is -180 to +180.

`lim:`*`datatype=low,high`* Sets the editing limits for a certain data type. The ***`datatype`*** is a number between 0 and 99, explained in Section 3.3. The arguments ***`low`*** and ***`high`*** overrule the limits set in the namelists by `limits(1,`*`datatype`*`)` and `limits(2,`*`datatype`*`)`. Example:

`lim:19=0,20`	Set the editing limits of wind speed to 0 to 20 m/s

`h=`*`low,high`* Sets the editing limits for sea level anomaly. It overrules the limits set in the namelists by `limits(1,0)` and `limits(2,0)`. The following examples have the same effect:

`h=-10,10`	Set the editing limits of sea level anomaly to -10 to +10 m
`lim:0=-10,10`	(same as above)

`sel=`*`sel1`*`[,`*`sel2`*`[,...]]` Selects one or more data fields for output. The numbers ***`sel1`***, ***`sel2`***, *etc.*, are the data field descriptors explained in Section 3.3. The numbers can be in the range 0-99 or 100-9999. Several **`sel=`** arguments may appear on the command line. Examples:

`sel=0`	Output sea level anomaly only
`sel=12,1207,2`	Output default and GOT00.2 ocean tide and latitude
`sel=12 sel=1207,2`	(same as above)

`opt=`*`descriptor`*`[,...]` Overrules the option number specified in the namelists. For example:

`opt=1207`	Make GOT00.2 the default ocean tide

Each **`opt=`** argument can contain one or more descriptors. Also, **`opt=`** arguments can be repeated. Hence, the following combinations of **`opt=`** arguments are equivalent:

`opt=1207,1307,1601`	Make GOT00.2 the default ocean and load tide
`opt=1207,1307 opt=1601`	and EGM96 the default reference surface

`fact:`*`datatype=factor`* Changes the multiplication factor for a certain data type in the construction of the sea level anomaly. The ***`datatype`*** is a number between 0 and 99, explained in Section 3.3. The argument ***`factor`*** overrules the value set in the namelists by `factors(`*`datatype`*`)`. Example:

`fact:15=0`	No longer account for sea state bias in the sea level anomaly

`debug=`*`level`* Sets the debugging level. The higher the level the more output is provided. Useful levels are:

`debug=0`	Most silent mode
`debug=1`	Provide some information on running process
`debug=2`	Level 2 and higher produces a lot of output

`-v` Increased the debugging level by 1. Using one **`-v`** is the same as using **`debug=1`**; **`-v -v`** is equivalent to **`debug=2`**.

4.2 rads2asc

The program rads2asc lists a selection of the RADS data base in one ASCII file or in several pass-by-pass ASCII files. These files contain a header per pass with a description of the data content followed by one record for each measurement that passes the selection criteria. The records are build up of space-separated columns listing various data fields. The first three columns are always time from the equator crossing (in seconds), latitude (in degrees), and longitude (in degrees). Which data fields are listed in addition to these three (and in which order they are listed) is fully configurable by the command line argument **`sel=`**.

In the output, the data that has not passed the editing criteria will be represented by "NaN" (Not-a-Number). If the sea level anomaly field suffers that fate, the record will not be listed in the output file, unless the **`-r`** option is used. In that case (with the **`-r`** option) the column associated with the sea level anomaly (when requested) will say "NaN" when invalid. Use **`-r#`** where **`#`** is a column number to specify another column that is used to eliminate data lines. Finally, one can also use **`-rn`** to indicate that a line should not be printed when any value is NaN.

It the argument **`out=`*`filename`*** is used, the output will go into one file (named ***`filename`***). Otherwise, pass files are created with the names `SSpPPPPcCCC.asc`, similar to what is described in Chapter 1.

Syntax

`rads2asc` ***`data set specifiers`*** **`[`*`data selection arguments`*`]`**

Data set specifiers

The required argument **`sat=`**, and the optional arguments **`cycle=`**, and **`pass=`** are described in Section 4.1.1.

Data selection arguments

The optional data selection arguments are described in Section 4.1.2. Those that are given a slightly different meaning in rads2asc and additional optional parameters are listed below.

`out=`*`filename`* Specify the name of the ASCII output file. When omitted pass files are created with names `SSpPPPPcCCC.asc`.

`sel=`*`sel1`*`[,`*`sel2`*`[,...]]` Specify the data field descriptors of columns 4, 5, *etc.* in the output. When omitted, the sea level anomaly, significant wave height and wind speed will be listed (in addition to time, latitude and longitude). In other words, the default is **`sel=0,17,19`**. With the **`-f`** option time, latitude and longitude will not be printed. See Section 4.1.2 for an extensive description of the **`sel=`** argument.

`-f` Do not print time, latitude and longitude columns, unless specified in the **`sel=`** option.

`-r` Print one record per measurement, also for rejected measurements. If omitted, the program only outputs the records in which the sea level anomaly (when requested in the output) passes the edit criteria.

-r# Print only those records for which column **#** is available and passes the edit criteria, i.e. it is not "NaN".

-rn Print only those records for which none of the columns are NaN.

-v Print a status bar on standard output. It will show the progress of the data selection.

Example

Assume you have installed a file `getraw.nml` as described in Section 3.5. Then issue the command

```
rads2asc sat=e2 cycle=0 pass=901,1000,2 out=output.asc sel=0,17,19 -v
```

The program will print to standard output information on which passes are available and have valid data points in the requested area (Mediterranean Sea), how many records were read and how many remained in the particular area, how many records were rejected based on the selection criteria on the corrections, and finally some statistics on the columns that were requested.

```
********************************************************************************

Data selection for satellite ERS-2 phase a

- = pass file does not exist
x = pass has no data in period and area
o = pass has no valid data
# = pass has valid data

Cycle Pass  ....+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8....+....9....+....0
    0  901  x x x x x x o o o x x x x x x x x x x x x o o x x x x x x x x x x x x o o x x - - - - - x x x x x o
###################################################################################################################
# Editing statistics for ERS-2
#
#     82328  Measurements read
#
#     81477  REJECTED:                                    ELEM FACT         MIN         MAX
#         0  time [yymmdd/hhmm]                            101  0.0        none        none
#     10533  latitude [degrees_north]                      201  0.0      28.000      48.000
#     70944  longitude [degrees_east]                      301  0.0      -8.000      42.000
#
#       851  Measurements in requested period and area
#
#       851  REJECTED:                                    ELEM FACT         MIN         MAX
#         0  orbital altitude, DGM-E04 [m]                 402  1.0  750000.000  850000.000
#         0  altimeter range corrected for instr. effect  601 -1.0  750000.000  850000.000
#         0  ECMWF dry tropospheric correction [m]         701 -1.0      -2.400      -2.100
#        11  wet tropospheric correction, MWR (NN) [m]     801 -1.0      -0.600      -0.000
#         0  alias: ionospheric correction, NIC09 [m]      906 -1.0      -0.400       0.040
#         0  total inverse barometer correction, MOG2D [ 1004 -1.0      -1.000       1.000
#         0  solid earth tide [m]                         1101 -1.0      -1.000       1.000
#       851  ocean tide [m]                               1207 -1.0      -5.000       5.000
#       851  load tide [m]                                1307 -1.0      -0.500       0.500
#         0  pole tide [m]                                1401 -1.0      -0.100       0.100
#         0  mean sea surface height, DTU10 [m]           1614 -1.0    -200.000     200.000
#        16  significant wave height [m]                  1701  0.0       0.000       8.000
#         8  backscatter coefficient [dB]                 1801  0.0       6.000      27.000
#        66  std dev of range (20-Hz) [m]                 2002  0.0       0.000       0.400
#        64  number of averaged 20-Hz measurements        2101  0.0      16.500      20.500
#       286  engineering flags                            2601  0.0   65512.000       0.000
#        63  std dev of significant wave height (20-Hz)  2802  0.0       0.000       2.100
#         0  off-nadir angle squared from waveform (smoo 3002  0.0      -0.050       0.050
#         0  reference frame offset [m]                   3801 -1.0      -1.000       1.000
#       851  sea level anomaly [m]                           0  1.0      -5.000       5.000
#
#         0  Valid sea level anomalies remaining
#
# Statistics of the requested data items
#
#    Number                                               ELEM FACT         MIN         MAX        MEAN      STDDEV
#       835  significant wave height [m]                  1701  0.0       0.000       7.960       1.487       1.045
#       849  wind speed [m/s]                             1901  0.0       0.000      24.944       5.775       3.433
###################################################################################################################
```

The resulting file `output.asc` will look like this:

```
...........  lines removed .............................
```

4.3 rads2grd

The program rads2grd is a quick-and-dirty gridding program for RADS data. No smoothing or interpolation is performed. The data are simply collected in cells of predefined size, after which their mean and rms-about-mean is computed.

The grid program not only grid sea level anomaly in longitude-latitude space, but any data field in any other space specified by the **sel=** option. For example, a non-parametric sea state bias model can be created by gridding the sea level anomalies in wind-wave space, in which case **sel=19,17,0** is used.

The boundaries of the grid are specified by the limits set in the `getraw.nml` namelists, or by the appropriate common command line arguments **lim=**, **lat=**, or **lon=**. The cell size is determined by the option **res=**. Alternatively, the options **x=** and **y=** can be used to set both the range and interval along both coordinate axis of the grid. The use of grid-node oriented or cell oriented boundaries is controlled by the **-C** option. The required minimum number of measurements per cell can be specified with the **min=** option.

The output is a list of ASCII records per cell: x and y of the centre of the cell, mean of z, rms-about-mean of z, number of points. Cells with a number of points less than the required number are not listed. This output can be used directly in GMT. In addition, for DEOS use, the options **grid=**, **mean=**, **rms=**, and **num=** can create DEOS grids.

Syntax

rads2grd *data set specifiers* [*data selection arguments*]

Data set specifiers

The required argument **sat=**, and the optional arguments **cycle=**, and **pass=** are described in Section 4.1.1.

Data selection arguments

The optional data selection arguments are described in Section 4.1.2. Those that are given a slightly different meaning in rads2grd and additional optional parameters are listed below.

sel=*xsel*,*ysel*[,*zsel*] Specify the data fields for the x-, y- and z-coordinate, where the z-coordinate is the actual data field to be gridded. When omitted, the sea level anomaly is gridded against longitude and latitude. In other words, the default is **sel=3,2,0**. See Section 4.1.2 for an extensive description of the **sel=** argument.

res=*xres*,*yres* Size of the grid cells in x- and y-direction. Default is **res=1,1**.

-C Normally, the boundaries are the centres of the outer cells. With the **-C** option, the boundaries will be regarded as the boundaries of the outer cells.

min=*nr* Specify the minimum number of points that have to fall in a cell in order to compute and output mean and rms-about-mean values. Default is **min=2**.

-v Print a status bar on standard output. It will show the progress of the data selection.

mean=*filename* Specify name for optional DEOS grid of mean values.

rms=*filename* Specify name for optional DEOS grid of rms-about-mean values.

num=*filename* Specify name for optional DEOS grid of number of values.

grid=*prefix* Equivalent to **mean=*prefix*_mean.grd rms=*prefix*_rms.grd num=*prefix*_num.grd**.

Example

Assume you have installed a file `getraw.nml` as described in Section 3.5. Then issue the command

```
rads2grd sat=e2 cycle=0 pass=901,1000,2 sel=19,17,0 -C -v
```

The program will print to standard output the same information as rads2asc (Section 4.2), followed by:

```
................ lines removed ...............
#    Number                                             ELEM FACT         MIN         MAX        MEAN      STDDEV
#       835  significant wave height [m]                1701  0.0       0.000       7.960       1.487       1.045
#       849  wind speed [m/s]                           1901  0.0       0.000      24.944       5.775       3.433
#################################################################################################################
```

4.4 radscolin

The program radscolin creates collinear tracks (in a rather straightforward way). No smoothing or interpolation is performed. The data are “gridded” based on their time with respect to the equator passages. The measurements are simply collected in bins of predefined length (1 second by default), after which the data falling in those bins for each specified pass are listed in a way similar to rads2asc (Section 4.2). No averaging or differencing is performed.

The program will output only those bins in which data can be found for all specified repeat cycles, unless the **`-r`** option is used.

The output is a list of ASCII records per bin, containing the data fields selected by the **`sel=`** option, for each of the specified cycles. In the current implementation 20 data fields from 400 different cycles can be compared (while the product of the number of data fields and the number of tracks can be no more than 2000). In contrast to rads2asc the time, latitude and longitude are not listed, unless requested with the **`sel=`** option.

Syntax

`radscolin` ***`data set specifiers`*** **`[`*****`data selection arguments`*****`]`**

Data set specifiers

The data set specifiers (**`sat=`**, **`cycle=`** and **`pass=`**) have the same meaning as described in Section 4.1.1.

`sat=`***`altimeter`*****`[:`*****`phase`*****`],[cycle=`*****`first`*****`[,`*****`last`*****`]`** and [`pass=`***`first`***`[,`***`last`***`]]]`

These arguments come in sets for each satellite, except for the **`pass=`** option, which applies to all satellites. If collinear tracks are selected from a single satellite (within the same mission) one set of **`sat=`**, with the optional **`cycle=`** and/or **`pass=`** is sufficient; for multi-satellite collinear tracks one additional pair of **`sat=`** and **`cycle=`** specifiers is needed for each satellite. Just as in the case of rads2asc, if the **`pass=`** argument is omitted, all passes within the cycles will used. If the **`cycle=`** argument is omitted, all cycles for that satellite (and mission) will be process Examples:

`sat=e2 cycle=0,15`	Print all collinear passes in ERS-2 cycles 0-15
`sat=e2 pass=57,302`	Print collinear passes 57 through 302 in all ERS-2 cycles

`sat=e2 cycle=77,79 sat=n1 cycle=10,12`
Print collinears of ERS-2 cycles 77-79 and Envisat cycles 10-12

Data selection arguments

The optional data selection arguments are described in Section 4.1.2. Those that are given a slightly different meaning in radscolin and additional optional parameters are listed below.

dt=*binsize* Specify the minimum bin size (in seconds). Default is **dt=0.97**.

-r Normally, only records are printed in which sea level anomalies are available for all cycles. With the **-r** option, all records are printed in which at least one cycle has the required data.

-r# Reject the data when there are less than **#** tracks with sea level anomalies. By default **#** is the total number of selected cycles.

-v Has no effect.

When you have not included sea level anomalies (**0**) as one of the items after **sel=**, radscolin will reject data based on the first data item specified.

Example

Assume you have installed a file `getraw.nml` as described in Section 3.5. Then issue the command

```
radscolin sat=e2 cycle=0,2 pass=915,1000,2 sel=2,0,17,19 -r
```

The program will print to standard output latitude, sea level anomaly, significant wave height and wind speed for all odd passes between 915 and 999 and cycles 0, 1 and 2. Each record will show these four quantities for each of the three cycles. With the **-r** option also records with invalid sea level anomalies are printed, as shown here:

```
# Satellite data selections:
#
#   1: sat=e2   cycle=000 pass=0915
#   2: sat=e2   cycle=001 pass=0915
#   3: sat=e2   cycle=002 pass=0915
#
# Columns   1 -   3 : latitude [degrees_north]
# Columns   4 -   6 : sea level anomaly [m]
# Columns   7 -   9 : significant wave height [m]
# Columns  10 -  12 : wind speed [m/s]
#
 30.185412        NaN  30.215553       NaN       NaN       NaN  3.740    NaN  6.140 21.129    NaN  0.728
       NaN  30.418223  30.459773       NaN       NaN       NaN    NaN    NaN  1.680    NaN  2.143  0.000
 30.487462  30.476067  30.518135       NaN       NaN       NaN  0.770  0.480  0.960  0.114  5.411  5.237
 30.545822  30.533894  30.575961       NaN       NaN       NaN  0.700  0.770  1.060  1.324  4.635  4.986
 30.603647  30.591735  30.633801       NaN       NaN       NaN  0.250  0.820  0.700  1.368  4.456  4.819
 30.661485  30.649574  30.691639       NaN       NaN       NaN  0.120  0.640  0.840  1.737  4.634  5.064
 30.719321  30.707412  30.749460       NaN       NaN       NaN  0.000  0.880  0.760  1.815  4.506  4.681
......................... lines removed ............................
       NaN  47.161520        NaN       NaN       NaN       NaN    NaN  0.250    NaN    NaN  3.544    NaN
# avg:  43.452607  43.447553  43.349090       NaN       NaN       NaN  1.205  0.638  0.749  6.318  2.805  1.221    0  999
# rms:   1.870872   1.893480   1.890970       NaN       NaN       NaN  0.858  1.033  0.594  3.565  1.992  0.825    0  999
# nr :      74     75     66      0      0      0     74     73     66     74     75     66      0   999
```

When the **-r** is omitted there will be no records with sea level anomalies equal to NaN. That means all cycles ought to have data, which is highly restrictive. The output will then be:

```
......................... lines removed ............................
```

To get all records in which *at least one* cycle has valid sea level anomalies, use the argument **-r1** on the command line.

4.5 max2

The program max2 is a crossover generating program, based on the age-old max program. Due to this heritage max2 has some archaic command line arguments and produces some historical binary formats. However, here only the arguments and options relevant to most RADS users are described. This includes the provision of an output file that can be easily read and is configurable.

The crossover generator can take a long sequence of cycles and create crossovers for all crossing passes. However, it is most efficient when the time period between the passes is limited to half the length of the repeat cycle. To limit the time interval, create a file `max.nml` as follows:

```
&nml
dtxo(4,4)=17.5,
dtxo(4,5)=4.9578,
dtxo(4,6)=4.9578,
dtxo(4,7)=17.5,
dtxo(5,5)=4.9578,
dtxo(5,6)=9.9156,
dtxo(5,7)=4.9578,
dtxo(6,6)=4.9578,
dtxo(6,7)=4.9578,
dtxo(7,7)=17.5,
dtxo(8,8)=8.525,
&end
```

The indices of `dtxo(i,j)` indicate the satellite numbers given in Table 3.1. This file will overrule any settings in the system-wide namelist `$RADSDATAROOT/nml/max.nml`. Other information essential to max2 comes from the namelist `$RADSDATAROOT/nml/satcat.nml`.

The output file will list the latitude and longitude of the crossover plus the time of the two crossing passes and any additional data field for each of the passes. For single-satellite crossovers first the interpolated value of the data field on the ascending pass is listed, then the value on the descending pass. For dual-satellite crossovers the values are listed for each satellite in the order as the **sat=** options appear. When the **-d** option is used, not the value for each pass, but the difference is provided as output.

It is important to realise that the difference of, for example, wind speed may suddenly be negative, whereas the data field itself normally is positive. This might lead to problems in the output, since max2 uses the output formats specified in the general and satellite-specific `getraw.nml` namelists, which may not have allowed for negative numbers to appear in such data fields. In the case of the wind speed, using the `f7.3` format, a number up to `99.999` can be printed out, but any value beyond `-9.999` will be printed as asterisk. You can avoid the problem by increasing the size of the format specifier to `f8.3`, as explained in Section 3.5.

Syntax

max2 ***data set specifiers*** **[*data selection arguments*]** ***prefix***

for large jobs (particularly dual-satellite jobs and jobs with a multitude of requested data fields) there is a version with more memory: max2_big.

Data set specifiers

The data set specifiers (**sat=**, **cycle=** and **pass=**) have the same meaning as described in Section 4.1.1, except that the ***step*** argument of **pass=** is ignored. One additional (and required) argument (***prefix***) serves as the prefix to the output file names.

sat=*altimeter*[:*phase*],[cycle=*first*[,*last*] and [
pass=***first***[,***last***]]]

These arguments come in sets for each satellite. If only single-satellite crossovers (within the same mission) are created one set of **`sat=`**, with the optional **`cycle=`** and/or **`pass=`** is sufficient; for dual-satellite crossovers two sets of these parameters are needed. Just as in the case of rads2asc, if the **`pass=`** argument is omitted, all passes within the cycles will used. If the **`cycle=`** argument is omitted, all cycles for that satellite (and mission) will be process Examples:

`sat=e2 cycle=1`	Make crossovers within ERS-2 cycle 1
`sat=e2 cycle=0,15`	Make crossovers spanning ERS-2 cycles 0-15
`sat=e2 pass=57,302`	Make crossovers from passes 57 through 302 in all ERS-2 cycles

`sat=e2 pass=57 sat=e2 pass=302`
Make crossovers at the crossing of passes 57 and 302 in all ERS-2 cycles

`sat=e2 cycle=0,15 sat=tx cycle=96,151`
Make crossovers between ERS-2 cycles 0-15 and TOPEX cycles 96-151

`prefix` is the prefix to all output file names. For each of the output files, a different extension will be added to the prefix. In the general case presented here, the only output file is `*prefix*.rxf`.

Data selection arguments

The optional data selection arguments are described in Section 4.1.2. Those that are given a slightly different meaning in max2 and additional optional parameters are listed below.

`-v` Has no effect.

`sel=`*`sel1`*`[,`*`sel2`*`[,...]]` Specifies up to 10 data fields that will be included in the output files. When omitted, **`sel=0`** is assumed. For more information on the **`sel=`** argument, see Section 4.1.2.

`-x2` Same as **`sel=0,17,19`**.

`-x3` Same as **`sel=0,17,10,5`**.

`-sA` Generate both single- and dual-satellite crossovers (default).

`-sS` Generate only single-satellite crossovers.

`-sD` Generate only dual-satellite crossovers.

`-d` List differences on crossovers, not both values.

Example

Assume you have installed a file `getraw.nml` as described in Section 3.5 and the `max.nml` file described above. Then issue the command

```
max2 sat=e2 cycle=0,1 sel=0,1,17 xovers
```

The program will print to standard output the following information:

```
 npar=           5
 ******************************************************************************
 *                                   MAX2 0309.0                              *
 ******************************************************************************

Options
   Maximum time interval over 6 observations   ->     12.000 sec.
   Process specifier                           ->      AQ-R-
   Short track orbit model specifier           ->      XXXXX
   Short track length                          ->          0 km.
   Minimum angle between tracks                ->      0.500 deg

Output files
   Altimeter file       -> -
   Crossover file       -> xovers.rxf
   Tracks catalogue     -> -

Reading RADS pass files ...

   Passes to be read    -> sat=e2   cycle=000,001 pass=0001,1002

                   2004   2%  .    .    .    .   50%   .    .    .    .  100%
           Reading passes ##################################################

   Number of measurements read                ->    2315077
   Number of measurements written             ->          0
   Number of tracks encountered               ->       1298
   Mean track length                          ->      18346 km.
   Percentage of short tracks                 ->          0 %


Guessing and determining crossovers ...

                 421197   2%  .    .    .    .   50%   .    .    .    .  100%
     e2 single-sat xovers ##################################################

   Track combinations analysed                ->     421197

   - Time interval out of limit               ->     293947
   - No intersection                          ->          0
   - Shallow angle                            ->          0
   - Too few data around                      ->     127250
   - Too many iterations required             ->          0

   Nr of xovers found and processed           ->          0
   Average number of track readings           ->      0.995
   RMS crossover time correction on guess [s]  ->        NaN
   Max. crossover time correction on guess [s] ->      0.000
   Crossover difference mean, RMS, stddev [cm] ->        NaN      NaN      NaN
   No xovers found
##################################################################################################################
# Editing statistics for ERS-2
#
#   2305009  Measurements read
#
#   2283256  REJECTED:                                   ELEM FACT         MIN         MAX
#         0  time [yymmdd/hhmm]                           101  0.0        none        none
#    294893  latitude [deg]                               201  0.0      28.000      48.000
#   1988363  longitude [deg]                              301  0.0      -8.000      42.000
#
#     21753  Measurements in requested period and area
#
#     21753  REJECTED:                                   ELEM FACT         MIN         MAX
#         0  alias: orbital altitude, D-PAF [m]           402  1.0  750000.000  850000.000
#         0  corrected altimeter range [m]                601 -1.0  750000.000  850000.000
#         0  dry tropospheric correction [m]              701 -1.0      -2.400      -2.100
#       510  alias: ECMWF wet tropospheric corrections [  801 -1.0      -0.600      -0.000
#         0  alias: ionospheric correction, NIC09 [m]     906 -1.0      -0.400       0.040
#         0  total inverse barometer correction, MOK2D [ 1004 -1.0      -1.000       1.000
#         0  solid earth tide [m]                        1101 -1.0      -1.000       1.000
#     21753  ocean tide [m]                              1207 -1.0      -5.000       5.000
#     21753  load tide [m]                               1307 -1.0      -0.500       0.500
#         0  pole tide [m]                               1401 -1.0      -0.100       0.100
#         0  geoid or mss height [m]                     1614 -1.0    -200.000     200.000
#       450  significant wave height [m]                 1701  0.0       0.000       8.000
#       203  backscatter coefficient [dB]                1801  0.0       6.000      27.000
#      1800  std dev of range [m]                        2002  0.0       0.000       0.400
#      1648  number of valid measurements                2101  0.0      16.500      20.500
#      7974  flag word                                   2601  0.0   65512.000       0.000
#      1850  std dev of significant wave height [m]      2802  0.0       0.000       2.100
#         0  off-nadir angle squared from waveform (smoo 3002  0.0      -0.050       0.050
#         0  reference frame offset [m]                  3801 -1.0      -1.000       1.000
#     21753  sea level anomaly [m]                          0  1.0      -5.000       5.000
#
#         0  Valid sea level anomalies remaining
#
# Statistics of the requested data items
#
#    Number                                              ELEM FACT         MIN         MAX        MEAN      STDDEV
#     21753  time [s]                                     101  0.0************************************ 1210996.876
#     21753  alias: orbital altitude, D-PAF [m]           402  1.0  787293.938  791131.450  789143.796     849.924
#     21303  significant wave height [m]                 1701  0.0       0.000       7.970       0.984       1.010
##################################################################################################################
```

And the crossover file `xovers.rxf` will contain the following lines.

```
# e2 single-sat xovers: asc-des
# Column  1 : latitude [deg]
# Column  2 : longitude [deg]
# Columns  3 -  4 : sea level anomaly [m]
# Columns  5 -  6 : time [s]
# Columns  7 -  8 : significant wave height [m]
#
............................. lines removed ..............................................
# Columns  3 -  4 : sea level anomaly [m]
# Columns  5 -  6 : time [s]
# Columns  7 -  8 : significant wave height [m]
#
```

Chapter 5

Creating your own program to read RADS data

Before other GETRAW routines can be used, GETRAW_INIT must be called to specify the satellite. Several alternative satellite names (specified in a namelist) are possible. The routine also loads default values for several options. These default values are specified in a general namelist $RADSDATAROOT/nml/getraw.nml and in satellite- and/or mission-specific namelists $RADSDATAROOT/nml/getraw_SS.nml. See Table 3.2 for a full list of these options. The values in these namelists can also be overruled or augmented by files getraw.nml and getraw_SS.nml in the working directory.

Optionally, one can call GETRAW_LIMITS and/or GETRAW_OPTIONS to change the limits for data editing and the preferred flavours of the various data fields.

Finally, GETRAW loads a single altimeter pass into memory and then returns the data values in SI units. Some selection codes do not refer directly to a data stored in the data files, but refer to functions of one or more data fields. Most notorious, of course, is the sea level anomaly, that is created as a linear combination of a satellite altitude, altimeter range and a number of geophysical corrections. All the requested data (including possible additional data needed to construct the data) is quality checked using the default limits or those specified by GETRAW_LIMITS. GETRAW_OPTIONS and GETRAW_FACTORS can be used to modify the construction of the sea level anomaly.

As a generic altimeter data user you do not want to do anything else than calling GETRAW. However, before the first call to this subroutine, one should use GETRAW_INIT to specify the satellite. The user can also enhance the data selection by adding one or more calls to GETRAW_OPTIONS, GETRAW_FACTORS and/or GETRAW_LIMITS.

Finally, GETRAW_STAT can be called to print out the statistics of the data reading and editing.

The following example will retrieve time, latitude, longitude, fully corrected sea level anomaly, and mean sea surface height from ten passes out of the data base. Only data over the Southern hemisphere is returned.

```
      integer maxdata
      parameter (maxdata=5000)
      real*8 eqtime,eqlon
      real*8 data(maxdata,2),time(maxdata),dlon(maxdata),dlat(maxdata)
      integer*4 cycle,pass,select(2),ndata,verbose
      character*256 mission,metafile

      mission = 'e1/d'
```

```
 verbose = 2
 select(1) = 0
 select(2) = 16

 call getraw_init(mission,verbose)
 call getraw_limits(2,-90d0,0d0)
 call getraw_options(16,2)

 do pass = 1,10
   call getraw (cycle,pass,2,maxdata,select,
|  time,dlat,dlon,data,ndata,eqtime,eqlon,metafile)

   do i = 1,ndata
     write(*,*) i,time(i),dlat(i),dlon(i),data(i,1),data(i,2)
   enddo
 enddo

 call getraw_stat(0)
 end
```

Appendix A
RADS Subroutine Descriptions

For a full description of the use of each of the routines, consult the subroutine descriptions below.

A.1 The implementation of GETRAW

A.1.1 GETRAW.INIT – Initalize GETRAW

```
    SUBROUTINE GETRAW_INIT (MISSION, VERBOSE)
    implicit none
    CHARACTER*(*) MISSION
    INTEGER*4     VERBOSE

This subroutine initializes the use of GETRAW. It MUST proceed any other
use of GETRAW, GETRAW_OPTIONS, GETRAW_FACTORS or GETRAW_LIMITS.
The first argument specifies the name of the altimeter MISSION; this
is a combination of the name of the satellite (or the altimeter) and
the mission phase. Many alternative satellite (altimeter) names can be
used. Examples: g3, GEOS-3, gs, GEOSAT, e1, ers1, ERS-1, e2, ers2, ERS-2,
tx, topex, TOPEX, tp, T/P, pn, POSEIDON, g1, GFO-1,
j1, JASON-1, n1, ENVISAT
Note: the case of the string MISSION is ignored: all characters are
first converted to lowercase.

For satellites with various mission phases (like ERS-1), the mission
phase ('a', 'b', etc) can be added to the satellite name, seperated by
a colon or a slash. Example: 'e1/c'. When the phase is omitted, a default
phase is assumed (see satellite-specific namelist).

This routine also loads the general and satellite dependent namelists.
These namelists contain (a.o.) the recommended flavours (options) for
each of the data elements and the limits for the quality checks. The
values in the satellie-dependent namelist overrule those in the
general namelist.

Namelists are loaded in the following order:
$RADSDATAROOT/nml/getraw.nml
$RADSDATAROOT/nml/getraw_<sat>.nml
$RADSDATAROOT/nml/getraw_<sat><phase>.nml
~/.rads/getraw.nml
~/.rads/getraw_<sat>.nml
~/.rads/getraw_<sat><phase>.nml
getraw.nml
getraw_<sat>.nml
getraw_<sat><phase>.nml

Values in these namelists can also be overruled by using
GETRAW_OPTIONS and/or GETRAW_LIMITS.

The argument VERBOSE is an integer number, specifying the level of
'verbosivety'. 0 means no messages are printed while processing, 1
```

```
produces progress bars, 2 additionally gives all data selection
criteria, 3 and higher are debug levels. The higher the number,
the more debugging information is provided. All verbose infomation is
printed on standard output.

Input argument:
 MISSION : Name of the satellite (altimeter), optionally followed by a
           colon (or s slash) and the mission phase.
 VERBOSE : Verbose level: 0 = Be silent; 1 = print progress bars;
           2 = print selection criteria; 3 and higher = debug levels
```

A.1.2 GETRAW — Get a pass from the RADS data base

```
    SUBROUTINE GETRAW (CYCLE, PASS, NCOLS, MAXDATA, SELECT,
   | TIME, DLAT, DLON, DATA, NDATA, EQTIME, EQLON, METAFILE)
    implicit none
    INTEGER*4     CYCLE, PASS, NCOLS, MAXDATA, SELECT(NCOLS), NDATA
    REAL*8        TIME(MAXDATA), DLAT(MAXDATA), DLON(MAXDATA),
   |     DATA(MAXDATA,NCOLS), EQTIME, EQLON
    CHARACTER*(*) METAFILE
```

```
This subroutine is a general purpose subroutine to read and retrieve
'raw' altimeter data from a specified altimeter pass and cycle out of
the RADS data base Version 2.0.

As a generic altimeter data user you do not want to do anything else
than calling this subroutine. However, before the first call to this
subroutine, one should use GETRAW_INIT to specify the satellite.
The user can also enhance the data selection by adding one or more
calls to GETRAW_OPTIONS, GETRAW_FACTORS and/or GETRAW_LIMITS.

See the description of GETRAW_INIT how to select the satellite,
GETRAW_OPTIONS for options/flavours related to the selected data
elements, GETRAW_FACTORS for the contribution of data elements to the
sea level anomaly and GETRAW_LIMITS for editing criteria.

CYCLE and PASS specify the cycle and pass number of the pass that has
to be loaded into memory. SELECT is an array of dimension NCOLS that
contains the NCOLS selection codes for the data elements that have to
be retrieved, edited and returned in matrix DATA.

The selection codes SELECT can be either of the following:
- Zero (0), indicating that the mean sea level anomaly is to be
  computed based on the options and factors set by default or by
  GETRAW_FACTORS and GETRAW_OPTIONS.
- A number smaller than 100, indicating the data element type.
  For example 4 indicates the orbital altitude. The default option value
  or the one set by GETRAW_OPTIONS will determine which type of orbit
  is used (e.g. 1 refers to JGM-3).
- A number greater than 100, indicating a specific data selection codes.
  The number is build up as EEFF, where EE is the data element
  number and FF is the option/flavour number. E.g., use 4001 for the
  JGM-3 orbital altitude.
See the manual for the full list of selection codes.

DATA is a matrix of dimension (MAXDATA,NCOLS) which will return the data
values specified by the SELECT array.

The following example will retrieve time, latitude, longitude, fully
corrected sea level anomaly, and mean sea surface height from ten passes
out of the data base. Only data over the Southern hemisphere is returned.

   integer maxdata
   parameter (maxdata=3500)
   real*8 eqtime,eqlon
   real*8 data(maxdata,2),time(maxdata),dlon(maxdata),dlat(maxdata)
   integer*4 cycle,pass,select(2),ndata,verbose
   character*256 mission,metafile

   mission = 'e1/d'
   verbose = 2
   select(1) = 0
   select(2) = 16

   call getraw_init(mission,verbose)
   call getraw_limits(2,-90d0,0d0)
   call getraw_options(16,2)

   do pass = 1,10
     call getraw (cycle,pass,2,maxdata,select,
  |  time,dlat,dlon,data,ndata,eqtime,eqlon,metafile)

     do i = 1,ndata
       write(*,*) i,time(i),dlat(i),dlon(i),data(i,1),data(i,2)
     enddo
   enddo

   call getraw_stat(0)
   end

The subroutine GETRAW only returns the data inside the requested time
range and area. The remaining data are quality-checked against
predefined limits. When data fields do not meet the checks, a VOID
value (1.D+30) is returned. The default limits are set by namelist
files (see manual) or can be specified with the GETRAW_LIMITS
routine (see below).

Input arguments:

  CYCLE   : Cycle number
  PASS    : Pass number
  NCOLS   : Number of elements (columns) to be read (after time, lat, and
            lon); NCOLS should be less or equal to the dimension of
            SELECT and the second dimension of DATA as defined by the
            calling (sub)program.
  MAXDATA : Maximum number of data records (rows); this is the dimension
            of the arrays TIME, DLAT, DLON and the first dimension of
            DATA, as defined by the calling (sub)program.
  SELECT  : Selection codes of the data elements; at least NCOLS values
            should be set. Note that the selection code can also be
            a combination of data element and data flavour. See above.

Output arguments:

  TIME    : Array with the time stamps of the data records in seconds since
            1.0 January 1985 or since the closest equator crossing (EQTIME)
  DLAT    : Latitude of the data records in degrees
  DLON    : Longitude of the data records in degrees
  NDATA   : The number of data values (rows) retrieved or:
            -1 = no pass file for the given satellite/phase/cycle/pass
             0 = there is a pass file, but no data in the selected time/area
  DATA    : Data returned (depends on SELECT)
  EQTIME  : Equator time passage [sec from 1985.0]
  EQLON   : Equator longitude [deg]
  METAFILE: Full path name of the meta data file used by GETRAW.
```

A.1.3 GETRAW.OPTIONS – Select options for GETRAW data items

```
    SUBROUTINE GETRAW_OPTIONS (SELECT, OPTION)
    INTEGER*4 SELECT, OPTION

This subroutine supports GETRAW. It can be used to set selection
options for individual data items, as well as for the fully corrected
sea level anomaly.

Each data item is referred by a selection code (SELECT). For each
data item there are one or more selection options (OPTION). The
value of OPTION can be zero, negative or positive. The meaning of
SELECT and OPTION is explained below.

Input arguments:

SELECT is an integer value between 0 and 99 (see manual for a
description of each value).
```

```
OPTION is an integer value and can be 0, negative or positive.
1) OPTION = 0 : When the data item is requested by GETRAW, all values
                returned will be NaN.
                The data item will be ignored in the construction of
                the fully corrected sea level anomaly.
2) OPTION > 0 : When the data item is requested by GETRAW, the value
                related by +OPTION is returned (see manual) after
                checking against its limits (see GETRAW_LIMITS).
                The data item will be used/checked in the construction
                of the fully corrected sea level anomaly.
3) OPTION < 0 : When the data item is requested by GETRAW, the value
                related by -OPTION is returned (see manual) after
                checking against its limits (see GETRAW_LIMITS).
                The data item will be ignored in the construction of
                the fully corrected sea level anomaly.

Default values for OPTION are given in the manual. They are retrieved
at run time from $RADSDATAROOT/nml/getraw*.nml and ./getraw*.nml (see also
GETRAW_INIT and manual).
```

A.1.4 GETRAW.FACTORS – Select factors for GETRAW data items

```
    SUBROUTINE GETRAW_FACTORS (SELECT, FACTOR)
    INTEGER*4 SELECT
    REAL*8    FACTOR

This subroutine supports GETRAW. It can be used to set multiplication
factors to each of the data items that make up the fully corrected
sea level anomaly.

Each data item is referred by a selection code (SELECT). And each data
item can be used to make up the fully corrected sea level anomaly.
This FACTOR is usually either -1d0, 0d0, or 1d0, indicating that it
is either subtracted from the sea level, ignored, or added to the sea
level anomaly. However, selecting FACTOR=0d0 does not mean that the
data item is not used for screening the data; the data field will
only be ignored when OPTION=0.

Examples:
1) The latitude (SELECT=2) does not contribute to the sea level anomaly,
hence FACTOR=0d0. But it is important for the screening or selection
of the data, hence OPTION=1 (and not 0).
2) The orbital altitude (SELECT=4) adds to the sea level.
If you want the DGM-E04 orbit, you will have FACTOR=1d0 and OPTION=2.
3) The ocean depth is not important in the screening and does not
contribute to the sea level anomaly. However when requested, we want the
value to be returned: FACTOR=0d0 and OPTION=-1.
4) There is no sigma SWH for Poseidon (SELECT=28): OPTION=0 and
FACTOR is irrelevant.

Default values for FACTOR are given in the manual. They are retrieved
at run time from $RADSDATAROOT/nml/getraw*.nml and ./getraw*.nml (see also
GETRAW_INIT).
```

A.1.5 GETRAW.LIMITS – Select limits for quality checking in GETRAW

```
    SUBROUTINE GETRAW_LIMITS (SELECT, LO, HI)
    INTEGER*4 SELECT
    REAL*8    LO, HI

This subroutine supports GETRAW. It can be used to set selection
boundaries for individual data items, as well as for the fully corrected
sea level anomaly.

Each data item is referred by a selection code (SELECT). And each data
item can be limitted by the minimum value LO and maximum value HI.

When defining the quality checks for the flag-word (SELECT=26), LO and
HI have a special meaning. Data rejection on the basis of the flags
will occur when either: (FLAGS.AND.LO).NE.0 or when
(FLAGS.AND.HI).NE.HI. In other words, LO is a bit flag indicating
```

```
which flag bits SHOULD NOT be set, HI defines which flag bits SHOULD
be set.

Default values for LO and HI are given in the manual. They are retrieved
at run time from $RADSDATAROOT/nml/getraw*.nml and ./getraw*.nml (see also
GETRAW_INIT).
```

A.1.6 GETRAW.STAT – Report rejections statistics from GETRAW

```
    SUBROUTINE GETRAW_STAT (UNIT)
    INTEGER UNIT

This subroutine reports the rejection statistics from GETRAW and
reinitializes the statistics. Output is to unit number UNIT.

Argument:
 UNIT (input) : Unit number for output:
                0 = standard error
                6 = standard output
```

A.2 Supporting routines

A.2.1 RADSARGS1 – Scan arguments for required RADS data selectors

```
    FUNCTION RADSARGS1 (USAGE, SAT, DEBUG)
    LOGICAL*4 RADSARGS1
    INTEGER*4 USAGE, DEBUG
    CHARACTER*(*) SAT

This routine scans the argument list for data selectors that
are required by most RADS programs, such as:

 sat=sat[/phase]: specify satellite and phase
 -v             : increase verbose/debug level by one
 debug=level    : set verbose/debug level

where [] denote optional arguments.

This routine should be run BEFORE a call to GETRAW_INIT if you want
to have an easy time getting this basic information from the command
line. If you do not use these command line options, this subroutine
is not needed.

The argument USAGE can be used to print the "usage" information.
The routine will return the selected satellite and mission,
and the requested debug (or verbose) level.

The function returns .true. if the basic information (satellite,
cycle numbers and pass numbers) are not set.

Input argument:
 USAGE       : = 0, print no usage, even when error occurs
               = 1, print no usage, unless error occurs
               = 2, print usage in addition to scanning arguments
               = 3, print usage only, do not scan arguments

Output arguments and return value:
 SAT         : Satellite and mission
 DEBUG       : Debug/verbose level
 RADSARGS1   : .false. = proper input
               .true.  = insufficient or erroneous input
```

A.2.2 RADSARGS2 – Scan arguments for optional RADS data selectors

```
    FUNCTION RADSARGS2 (USAGE, CYC0, CYC1, PASS0, PASS1, DPASS,
   |                    NSEL, SEL)
    LOGICAL*4 RADSARGS2
    INTEGER*4 USAGE, CYC0, CYC1, PASS0, PASS1, DPASS,
   |          NSEL, SEL(NSEL)
```

```
This routine scans the command argument list for optional arguments
that are related to cycle range, pass range, selection codes,
data limits, options and factors used by RADS, such as:

 pass=p0[,p1[,dp]] : specify first and last pass and modulo
 cycle=c0[,c1]  : specify first and last cycle
 sel=sel0,...   : set selection codes
 + all options that RADSOPTION supports.

This routine should be used AFTER the call to GETRAW_INIT if
you want to have easy access to these command line options.
The routine will overrule the limits, options and factors that
are set in the default namelists.

The argument USAGE can be used to print the "usage" information.
The routine will return the selected cycle numbers, pass numbers,
and the selection codes. Set NSEL to 0 if you do not want to read
selection codes.

Input argument:
 USAGE     : = 0, print no usage, even when error occurs
             = 1, print no usage, unless error occurs
             = 2, print usage in addition to scanning arguments
             = 3, print usage only, do not scan arguments
               To this add:
                10, to ignore the order and range of cycle numbers
               100, to force the use of sel= option
 NSEL        : Maximum number of selection codes (use 0 if to be ignored)

Output arguments and return value:
 CYC0,CYC1   : First and last cycle number
 PASS0,PASS1 : First and last pass number
 DPASS       : Pass number interval
 NSEL        : Actual number of selection codes
 SEL         : List of selection codes
 RADSARGS2   : = .false., proper input
               = .true. , insufficient or erroneous input
```

Appendix B
Database Layout

This chapter is added for those who have the desire to understand how the data base is built. It is of little of no consequence if you do not read it, but may give you a better insight when you do.

RADS was first conceived in 1998. The data base had a format similar to most GDR products, in which the record length is fixed, and the order and specification of the various data fields is likewise unchangeable. There were several reasons to change the format of the RADS data base.

- The RADS v1.0 format is rather inflexible.
- The record length of 80 characters limits the extension with additional fields.
- Some satellites do not require all fields, hence disk space is used inefficiently.
- Testing extra or new geophysical corrections is difficult without impact for other users.
- Any change to the format requires a complete reproduction of the data and has similar impact to the programs that read the data base.

Just changing the format was **no option**. It also requires the coding and use of a new GETRAW routine. Reading the data directly with a C or FORTRAN `read` statement should be avoided. The format change was paired with the generation of a GETRAW routine that not only reads the data, but also converts it to the appropriate units, does error checking and data screening.

B.1 Advantages of the netCDF format

The netCDF format is extremely flexible, and platform independent. Here are some of the key elements that made us chose the netCDF format of the altimeter data.

Pass files: The idea of pass files is kept. It seems the best way to deal with altimeter data. The naming convention of the pass files is not much of a concern to the GETRAW subroutine, the format for this is fed by a namelist entry.

Meta data: The meta data is part of the data files. This makes the data files self-descriptive, so not much additional information is needed. This allows also other programs (outside RADS) to make sense of the data.

"Rotated" data file: The word "rotated" refers to the idea that the conventional GDR data is record oriented (one record contains a variety of different fields), while the netCDF files are column, or field oriented (each ”record” contains data collected on different times).

Order: The order in which the data fields are stored in the product has become irrelevant. Information about the field order is contained in the meta data.

Flexibility: The idea of fixed record lengths is left. Reading is no longer performed by fixed record length Fortran read statements, but by netCDF read statements. Although this requires extra coding, it is much more system-independent, and allows easy augmentation of the data files.

Byte representation: The netCDF library takes care of the issue of byte representation that makes it possible even to use single and double floats, not only integers.

B.2 Data structure

The RADS netCDF data files are filled field by field, instead of record by record. That looks strange at first, but in the end this provides the most flexible solution to add or remove fields at will. One may even use a similar format for SLA (sea level anomaly) files.

The conventional way to construct an altimeter product is record by record. A **record** is a combination of measurement and model information pertaining to a single time tag. These measurement and model values are called **fields**. A conventional altimeter product would have, for example, n data records of m fields each.

In the netCDF format the columns and rows in the data product are exchanged. The number of “records” is now equal to the number of different data fields (m). Each record contains the n values pertaining to each of the time tags. Since a data field can be either 1, 2, or 4 bytes long, the records are not of equal length. The data fields remain, preferably, of integer type. But floats can be used as well.

It is believed that in this way, the format of the data file is flexible enough to allow additions of data fields or remove data fields at will, without having impact on the user and his/her software.

The rotated way of data storage has an additional unforeseen advantage. Synchronising data sets can be sped up significantly using the program `rsync` (freeware, binaries available). This program can do synchronisation of files across the network while sending only the changes, and not the whole file. This means that when a field is added, simply the extra record is transferred and pasted into the remote file. In case of an update of a column, the old one will be removed and the new one is put in place.

To learn more about the netCDF files, see the UCAR netCDF web site.

B.3 Role of the meta data

The netCDF meta data contain sufficient information to “understand” the contents of the file. This is **more** than just specifying the number of records and data fields. It also specified the variable name, data unit, scale factor, offset, etc. All information that can be used directly by any software to convert the data in a humanly readable form.

B.4 A remnant of the past: the RMF file

Next to the netCDF data files, RADS still has a remnant of the past: the `getraw.rmf` files. These files were previously used as meta files but now only contain a few extra tricks to replace data fields or compute them on the fly. The RMF files are a compilation of **free-format** ASCII lines of mixed text and values. The start of the record (*i.e.*, the first word, or *key*) specifies the meaning of the line. When parsing the lines GETRAW simply ignores all lines that start with an unknown key. The key is recognised by its first 5 letters. The order of the lines is irrelevant.

Each key is followed by one or more values or character strings (within quotes). The values and strings are separated from the key and from each other by one or more spaces or tabs.

@RADS_RMF_V2.2 The first line specifies that this is a RADS RMF version 2.2 file

# or % Lines starting with `#` or `%` can be used to enter comments. They are ignored by GETRAW.

CONST indicates a constant. Obvious candidates of data types that are stored as constants are altimeter biases. But also slowly varying values that can be assumed constant over one pass can be stored as a constant. The key is followed by the data field descriptor (an integer number) followed by a string denoting the meaning of this field and the unit.

MATH specifies a function written as a sequence of operators on constants or data columns. The formulation is much like the GMT program gmtmath. The key (`MATH`) is followed by:

- The data field descriptor.
- A character string expressing the mathematical function.
- A string describing the function.

For a description of all mathematical operators see Chapter C.

ALIAS points to another field. Its arguments are:

- The data field descriptor.
- The data field descriptor the previous argument is pointing to.
- A character string that describes the alias.

UNDEF removes a previously defined field from memory. It has only one argument, the field descriptor of that has to be erased.

Appendix C
Math operators

The mathematical expressions that can be used on `MATH` records are very similar to the program `gmtmath` of the GMT plotting package. The expressions are parsed from left to right and are to be written in a Reverse Polish Notation. This means that, for example, $A + B =$ is written as `A B ADD =`. The expression always needs to end in an equal sign (=).

The expressions operate on column data, so in the previous example, all elements of the vector A is are added to the corresponding elements of vector B. The output is a new vector. To use data fields as vectors, use a dollar sign (`$`) followed by the field number. If the data is to be tested against the predefined limits, use a percent sign (`%`) followed by the field number. Examples: `$1207` and `%1207`.

A large number of operaturs can be used, as described in Table C.1. The functions take 0, 1, 2, or 3 columns as input, and return 1, 2, or 3 columns at the top of the stack. Table C.1 also indicates the number of input and output columns. Constants (like `E` and `PI`) and values (like `2` or `3.14d0`) do not take any input.

Operator	Input	Output	Description
ABS	1	1	abs (A)
ACOS	1	1	acos (A)
ACOSH	1	1	acosh (A)
ADD	2	1	A + B
AND	2	1	B if A == NaN, else A
ASIN	1	1	asin (A)
ASINH	1	1	asinh (A)
ATAN	1	1	atan (A)
ATAN2	2	1	atan2 (A, B)
ATANH	1	1	atanh (A)
AVG	2	1	A if B == NaN, B if A == NaN, else (A+B)/2
BTEST	2	1	btest(A, B)
CEIL	1	1	ceil (A) (smallest integer $\geq$ A)
COS	1	1	cos (A) (A in radians)
COSD	1	1	cos (A) (A in degrees)
COSH	1	1	cosh (A)
D2R	1	1	Converts Degrees to Radians
DETREND	2	1	Remove linear trend from B=f(A)
DIV	2	1	A / B
DUP	1	2	Places duplicate of A on the stack

Table C.1 Math operators that can be used in MATH records.

Operator	Input	Output	Description
E	0	1	Natural exponent = exp(1)
ERF	1	1	Error function erf (A)
ERFC	1	1	Complementary Error function erfc (A)
EQ	2	1	1 if A == B, else 0
EXCH	2	2	Exchanges A and B on the stack
EXP	1	1	exp (A)
FLIPUD	1	1	Reverse order of each column
FLOOR	1	1	floor (A) (greatest integer $\leq$ A)
FMOD	2	1	A % B (remainder)
GE	2	1	1 if A $\geq$ B, else 0
GT	2	1	1 if A $>$ B, else 0
HYPOT	2	1	hypot(A, B) = $\sqrt(A^2 + B^2)$
INT	1	1	Numerically integrate A
INV	1	1	1 / A
ISNAN	1	1	1 if A == NaN, else 0
LE	2	1	1 if A $\leq$ B, else 0
LOG	1	1	log (A) (natural log)
LOG10	1	1	log10 (A) (base 10)
LOG1P	1	1	log (1+A) (accurate for small A)
LOG2	1	1	log2 (A) (base 2)
LOWER	1	1	The lowest (minimum) value of A
LT	2	1	1 if A $<$ B, else 0
MAX	2	1	Maximum of A and B
MEAN	1	1	Mean value of A
MIN	2	1	Minimum of A and B
MUL	2	1	A * B
NAN	2	1	NaN if A == B, else A
NEG	1	1	-A
NEQ	2	1	1 if A != B, else 0
NRAND	2	1	Normal, random values with mean A and σ B
OR	2	1	NaN if A or B == NaN, else A
PI	0	1	π
POP	1	0	Delete top element from the stack
POW	2	1	A^B
R2	2	1	R2 = $A^2 + B^2$
R2D	1	1	Convert Radians to Degrees
RAND	2	1	Uniform random values between A and B
RINT	1	1	rint (A) (nearest integer)
SMTHIONO	2	3	Smoothes ionosphere correction (input: time, iono)
SIGN	1	1	sign (+1 or -1) of A
SIN	1	1	sin (A) (A in radians)
SINC	1	1	sinc (A) (sin (pi*A)/(pi*A))
SIND	1	1	sin (A) (A in degrees)
SINH	1	1	sinh (A)
SQRD	1	1	A*A
SQRT	1	1	sqrt (A)
STD	1	1	Standard deviation of A
STEP	1	1	Heaviside step function H(A)
STEPT	1	1	Heaviside step function H(t-A)
SUB	2	1	A - B
SUM	1	1	Cumulative sum of A

Table C.1 Math operators that can be used in MATH records (cont'd).

Operator	Input	Output	Description
TAN	1	1	tan (A) (A in radians)
TAND	1	1	tan (A) (A in degrees)
TANH	1	1	tanh (A)
TREND	2	1	Fit linear trend through B=f(A)
UPPER	1	1	The highest (maximum) value of A
XOR	2	1	B if A == NaN, else A
YMD	1	1	Convert SEC85 to YYMMDD.DDD
YMDHMS	1	1	Convert SEC85 to YYMMDDHHMMSS.SSS

Table C.1 Math operators that can be used in MATH records (cont'd).

Appendix *D*

Version History

D.1 Version 1, November 1998

RADS was first conceived in November 1998. RADS Version 1, as it is now referred to, was much less flexible than the current version. But it has the principle idea of providing a generic data base with a software interface that would do editing on the fly. However, unlike RADS Version 2, it did not allow any flexibility in the data content or format, nor in the way the editing was performed.

D.2 Version 2, December 2000

RADS Version 2 was developed in December 2000, and was a mayor improvement over its predecessor. It required the regeneration of the data base (which until then only included TOPEX/Poseidon, ERS-1 and ERS-2 data) and the development of new tools for the generation and manipulation of the data. This is the version as described in this user handbook.

But also RADS Version 2 has had its improvements over the years, of which the most important ones are listed in this and the next Sections.

- Data of Geosat, GFO, Jason-1, and Envisat are added to the data base.
- Provide also C- and S-band measurements for dual-frequency altimeters.
- TOPEX significant wave heights are patched for the period when the Side-A altimeter is deteriorating, such that they maintain, on average, a constant offset with the ERS-2 wave heights.
- FES99 and GOT00.2 ocean tides are added to all data sets.
- ETOPO2 topography is added to all data sets.
- CODE GIM, JPL GIM and IRI95 ionospheric delay corrections are added for all data since September 1998.
- Adding global mean ocean pressure to the meta files.
- Gradual adding of new functionalities, like on the fly application of operators on the data.
- Data can be created both in big or little endian format.

D.3 Version 2.1, June 2003

This is a mayor revision of the RADS data set and software. Although the old data base is compatible with the new software, and visa versa, we recommend to update both the software and the data base. Note that the new namelists do not work with the old software.

What is new in the software?

- Namelists specific to each altimeter mission phase can be used.
- User specific namelists (for general, satellite-specific or mission-specific settings) can be created in `~/.rads`. Same for `getraw.rmf` files.
- The **`pass=`** and **`cycle=`** command-line arguments need no longer be specified both. RADS now "knows" the number of passes in a cycle, and the number of cycles in the mission phase.
- The radsargs1 and radsargs2 have been better designed. Most arguments from radsargs1 are moved to radsargs2. *To use these new routines in the programs you created yourself, you need to change your code!*
- The arguments of radscolin are rearranged, similar to max2; radscolin also allows more tracks to be compared.
- On-the-fly interpolation of grids implemented.
- New organisation of source directories into utilities, tools, utilities under development, and supporting tools.
- Namelists are adjusted to reflect the new data holdings.
- Source, namelists and scripts are added to CVS tree.

What is new in the data base?

- The GOT00.1 and CLS01 mean sea surface models have been added to all data sets.
- The EGM96 geoid and GOT00.1 and CLS01 mean sea surface models are now interpolated using bi-cubic splines. This provides much better results, particularly for EGM96.
- The OSU MSS95 mean sea surface model (using bi-linear interpolation) is still available but will be depreciated in the near future.
- The routines for FES2002 and GOT00.2 have been improved to avoid errors at the sub-millimetre level.
- FES99 has been recomputed after fixing a bug in the FES tide prediction software.
- The CSR 3.0 and FES95.2.1 ocean tide models have been removed.
- The routines for solid earth tide and Modified Chelton and Wentz (wind) model have been improved to avoid errors at the sub-millimetre level.
- The drift in the ERS-2 microwave radiometer brightness temperature has been compensated.
- The GFZ and JGM-3 orbits on the ERS-2 products have been removed.
- Added IRI95 ionospheric delay correction to all data sets.
- Added JPL GIM ionospheric delay on all data sets since September 1998; removed the CODE GIM ionospheric delay.

What is new in the namelists (settings)?

- Special settings per altimeter mission phase can now be used.
- Namelists now include pass and cycle ranges.
- The following fields have become default:
  - Inverse barometer correction referenced to the global mean pressure over oceans (1003), instead of the correction that did not include the mean pressure correction (1001)
  - GOT00.2 ocean and load tide (1207, 1307) instead of CSR 3.0 tides (1201, 1301)
  - CLS01 mean sea surface as reference (1605) instead of OSU MSS95 (1602)
- For GFO and Geosat: extended number of averaged 10-Hz measurements from 9-10 to 8-10.

D.4 Version 2.2, December 2004

During the year several changes have been made to the data base as well as to the software. Because of the implementation of new record types in the meta files, the data base is not compatible with older software versions. Both the namelists and software needs to be upgraded to work with the latest implementation of the data base.

What is new in the software?

- Several minor bugs have been patched.
- Because of problems on SUN platforms with the recursive calling of the subroutine `LOAD_ANY`, some reworking of the code had to be undertaken. The redesign was trivial, except for `LOAD_FUNC`, which has now been depreciated and replaced by the much more flexible `LOAD_MATH`.
- The `ALIAS` record was introduced. This makes is easy to assign more than one field number to the same data.
- The `MATH` record was introduced. See description in this manual.
- Invalid data is now set to NaN (Not-A-Number) instead of 1.D+30. Use the function `ISNAN` to trap them.

What is new in the data base?

- Many fields have been added, particularly for TOPEX.
- GFO has now all PGS7777b orbits.
- FES2004 is replacing FES2002.
- Envisat data corrected for drifts.
- Geosat data undergoing reconstruction.
- New microwave radiometer drift corrections for ERS-1, ERS-2, Envisat and T/P.

D.5 Version 3.0, February 2008

What is new in the software?

- RMF and RDF files replaced by netCDF files.
- Provide better interface with GMT.

D.6 Version 3.1, June 2008

What is new in the software?

- Introduced configure to configure the software. This removes the `sysdep` directory and the system dependent include directories.
- Environment variable `RADSROOT` no longer used/necessary. `RADSDATAROOT` can be used to point to the data base.
- `nml` directory moved the `data/nml`.
- All scripts are installed from the `src` subdirectories, nothing remains in `bin`.
- When upgrading using CVS, please remove the directories `bin`, `lib`, and `src`. Then checkout `src` anew. Move drectory `nml` to the data directory.

  What is new in the data base?
- GGM02C ITRF2000 and ITRF2005 orbits for TOPEX and Jason-1.
- Range bias between TOPEX side A and B fixed.
- New tide models GOT 4.7, WebTide Arctic8d, WebTide Hudson.
- New surface models EGM2008 MSS, EGM2008 geoid, DNSC08 MSS, DNSC08 bathymetry.