Technical report; TR-279

KNMI

Sciareadlv1

a FORTRAN 90 interface to SCIAMACHY level-1c files

Jeroen van Gent and Pieter Valks

De Bilt, 2005

Technical report = technisch rapport; TR-279

De Bilt, 2005

PO Box 201
3730 AE De Bilt
Wilhelminalaan 10
De Bilt
The Netherlands
http://www.knmi.nl
Telephone +31(0)30-220 69 11
Telefax +31(0)30-221 04 07

Author: Gent, J. van
Valks, P.

● ● ● ●

Sciareadlv1

a FORTRAN 90 interface to SCIAMACHY level-1c files

Jeroen van Gent, Pieter Valks[1]

KNMI Royal Netherlands Meteorological Institute

[1] Currently at: Deutsches Zentrum für Luft- und Raumfahrt e.V., Oberpfaffenhofen, Germany

Sciareadlv1

1 Introduction

This document briefly describes the FORTRAN 90 software for the reading of datasets from level 1c files from the SCIAMACHY satellite instrument [1]. The main code components are described and a short example of the application of the software is given.

2 Sciamachy level1 files

SCIAMACHY data come as binary files, each generally containing one satellite orbit of data. In general, a user will receive geolocated, uncalibrated data, so called level 1b. Before this data can be applied for scientific use, it needs to be calibrated. This can be done by means of the SciaL1C tool, which is included in the ESA Enviview software, developed to view ENVISAT data.[1]

SciaL1C converts files of level 1b type to level 1c type. In both file types information is stored in the SCIAMACHY specific binary file format PDS. Soon after the launch of ENVISAT, the NADC library software was developed for reading PDS data into memory from C programs and IDL[2]. Based on the NADC library, in 2002 a FORTRAN 90 wrapper was developed at KNMI. This wrapper, called SCIAREADLV1, is the subject of this document.[3] Detailed information on SCIAMACHY data processing and calibration and the structure of PDS files can be found in the level 0 to level 1c Algorithm Theoretical Basis Document [2].

3 Sciareadlv1

SCIAREADLV1 is available at http://www.temis.nl/data/. The SCIAREADLV1 FORTRAN module is a FORTRAN 90 interface that calls C-routines for reading of PDS-data. The C-routines were derived directly from the NADC library, but were modified as to cooperate with FORTRAN. SCIAREADLV1 reads all SCIAMACHY data from a level 1c file and stores it into a derived type variable, from which it can be used for further processing.

4 Requirements

In order to use the SCIAREADLV1 software you need a Linux operating system or similar. SCIAREADLV1 has successfully been tested with Linux Red Hat 7.2 and Suse 9.1, but other Linux and Unix-type environments may also work. Furthermore, working FORTRAN 90 and C or C++ compilers are required.

[1] Alternative data processing methods are provided in the framework of the Netherlands Sciamachy Data Center (http://www.sron.nl/~hees/SciaDC/)

[2] This library is available at http://www.sron.nl/~hees/SciaDC/

[3] level 1c reading functionality with FORTRAN can also be found in the Basic Envisat Atmospheric Toolbox (http://www.science-and-technology.nl/beat/).

5 Installation

The SCIAREADLV1 routine comes as a gzip-ed tar-archive file (sciareadlv1.tar.gz). Copy the file to the location of your choice and type (> is the command prompt):

```
> tar -xvzf sciareadlv1.tar.gz
```

Enter the programs directory:

```
> cd sciareadlv1/
```

Here, among other files, you will find a makefile to compile the code and a `data` directory which contains an example of SCIAMACHY level 1c file.

In order for the SCIAREADLV1 to function with your fortran and C compilers, you have to modify the file called

```
makefile
```

Open this file in an editor. In the beginning of the file you will see two lines similar to:

```
FC    = pgf90
CC    = gcc
```

Replace pgf90 and gcc by your favorite FORTRAN and C compiler, respectively and save the file. Finally type:

```
> make
```

This creates the required object and module files to be used with your own software. Furthermore, an executable file, called `rundemo` is created, which we will use in the next section.

6 Using sciareadlv1

6.1 Calling the sciareadlv1 software

By examining the makefile, you should now be able to use SCIAREADLV1 with your own FORTRAN 90 software. Its use can best be explained by an example. Fig. 1 shows the content of the file `src\rundemo.f90`. The purpose of this little program is to read the content of the small SCIAMACHY level 1c file (to be found in the `data` directory) and print some data characteristics on the screen. It can be used as a basis for writing your own reading software.

The first two USE statements in the code are compulsory; they indicate the use of modules from the SCIAREADLV1software. In the TYPE statement that follows, the variable is defined in which the output is stored. Finally, the name of the level 1c file is required as well as the type of data of interest, in this case nadir data. After these preparations the actual call to the reading routine take place: `CALL ReadLevel1cData`... And finally, some of the retrieved data is printed on screen, to be examined next.

```
PROGRAM rundemo

  ! Used Modules
  USE defs_scia
  USE scialevel1cModule, ONLY : ReadLevel1cData

  TYPE(Level1DataType) :: Level1Data      ! Will contain the data
  CHARACTER(255)       :: Level1FileName  ! Level 1c file name
  INTEGER              :: Status          ! = 0 if reading succeeds
  LOGICAL              :: FileIsRead      ! = T if reading complete
  INTEGER              :: MDS_type
  INTEGER              :: istate, Nstates

  Level1FileName = './data/SampleLevel1File.l1c' ! Level 1c file
  MDS_type       = SCIA_NADIR       ! Type of data to be obtained

  ! read data
  CALL ReadLevel1cData                                 &
       ( Level1FileName, MDS_type, Level1Data,         &
       Status, FileIsRead, doreport=.TRUE. )

  ! show data
  WRITE(*,'(A,1X,I2)') 'Status:', Status
  WRITE(*,'(A,1X,L1)') 'Is file read?:', FileIsRead
  WRITE(*,*)

  IF ( FileIsRead ) THEN
     Nstates = SIZE(Level1data%mds_state)
     WRITE(*,'(A,A)') 'Product:',Level1data%mph%product
     WRITE(*,'(A,I3)') 'Total nr of states:',Level1data%dsd(1)%num_dsr
     WRITE(*,'(A,I3)') 'nr of selected states:',Nstates
     WRITE(*,*)
     WRITE(*,'(A8,2X,A13)') 'State ID','State Present'
     DO istate=1,Nstates
        WRITE(*,'(6X,I2,14X,L1)') Level1data%mds_state(istate)%state_id,&
           &              Level1data%StatePresent(istate)
     END DO
  END IF
  !
END PROGRAM rundemo
```

Figure 1: FORTRAN 90 example of using the SCIAREADLV1 software.

```
Opening Level1c Data file
Processing product:
SCI_NLC_1CNPDK20030521_200942_000011532016_00329_06395_1370.N1
 File will be searched for states of type: NADIR
 Total number of states (L1b): 16
 Total number of NADIR states (L1b):  8
 Selected number of states:   8
Closing Level1c Data file
Finished

Status:  0
Is file read?: T

Product:SCI_NLC_1CNPDK20030521_200942_000011532016_00329_06395_1370.N1
Total nr of states: 16
nr of selected states:  8

State ID  State Present
       1              T
       2              T
       3              T
       4              T
       5              T
       6              T
       6              T
       6              T
```

Figure 2: Screenoutput

6.2 Examining the output

To perform the actions of our example, execute the file rundemo, which was created if you followed the steps in section 5.

```
> rundemo
```

The output that appears on your screen is shown in Fig. 2. It shows that the level1c file is opened for reading. Among the first date retrieved from the file is the product name, in which information such as date-of-measurement and orbit number is contained. Subsequently, the program shows which type of data it will search for: nadir. In our example, a total of 16 files are indicated to be present in the data, of which 8 are of nadir type. Therefore, 8 states are selected for reading. Strictly speaking, these number indicate the amount of states as indicated in the header of the *level 1b* file. In general, they will match with the actual number of states present in the level 1c data, but incidentally, certain state data may appear not to be present or data may be corrupt. For that reason, while reading the level 1c file, state data is checked to be actually present.

From Figs. 1 and 2, we see that after the level 1c file has been read, some basic information is shown for each of the 8 selected nadir states. The state id number indicates the type of nadir measurement that was performed by SCIAMACHY and the StatePresent flag indicates that indeed all of the selected states were found to be present in the level 1c file.

7 Data structure

In the example program shown in Fig 1, the data read from the level 1c file is stored in a derived type variable named Level1data. The components of this variable and the available data fields can be examined on the following pages. For example: we can get the mean solar irradiance spectrum from the array: Level1cData%srs(1)%mean_sun, along with the corresponding wavelength grid in Level1cData%srs(1)%wvlen_sun.

The data field names are derived from their equivalents in the nadc library. The data storage hierarchy largely correspond to the way the data is stored in the level 1c files. Also use Enviview to find the data fields of your interest in your level 1c files. More information on how SCIAMACHY data is composed can be found in [2] and at the Sciamachy Operations Support (SOST) web site (http://atmos.af.op.dlr.de/projects/scops/).

7.1 Level1Datatype

structure Level1Datatype

Name	Type	Description
mph	type(mph_scia)	main product header
sph	type(sph1_scia)	specific product header
dsd	type(dsd_scia), dimension(:), pointer	data set description
sqads	type(sqads1_scia), dimension(:), pointer	summary of quality data per state
lads	type(lads_scia), dimension(:), pointer	geolocation of states
ppg	type(ppg_scia)	PPG/Etalon parameters
srs	type(srs_scia), dimension(:), pointer	sun reference spectrum
sfp	type(sfp_scia), dimension(:), pointer	slit function paramaters
mds_state	type(state1_scia), dimension(:), pointer	state data
calopt	type(cal_options)	calibration options
mds	type(mds1c_scia), dimension(:,:), pointer	measurement data set of one state
mds_pmd	type(mds1c_pmd), dimension(:), pointer	pmd mds of one state
mds_polV	type(mds1c_polV), dimension(:), pointer	fractional pmd mds of one state
StatePresent	LOGICAL, dimension(:), pointer	state data present indicator for level 1c file

7.2 Sciamachy Instrument specific parameters

Instrument constants

Name	Type	Value	Description
max_utc_string	integer	28	max Universal Time string length
science_channels	integer	8	number of science channels
pmd_number	integer	7	number of pmd channels
num_coords	integer	4	number of ground pixel coordinates
scia_nadir	integer	1	nadir measurement type flag
scia_limb	integer	2	limb measurement type flag
scia_occult	integer	3	occultation measurement type flag
scia_monitor	integer	4	monitoring measurement type flag
channel_size	integer	1024	number of pixels per channel
all_channels	integer	(science_channels + pmd_number)	Total number of channels
max_cluster	integer(kind=2)	64	max number of clusters per channel

7.3 Envisat data types

structure mph_scia

Name	Type	Description
leap_err	character(len=2)	leap second error
phase	character(len=2)	phase letter
proc_stage	character(len=2)	processing stage
product_err	character(len=2)	flag indicating if errors have been reported
product	character(len=63)	product file name
ref_doc	character(len=24)	reference document describing the product
acquis	character(len=21)	acquisition station ID
proc_center	character(len=7)	processing center ID
proc_time	character(len=max_utc_string)	UTC time of processing
soft_version	character(len=15)	software version number
sensing_start	character(len=max_utc_string)	UTC start time of data sensing
sensing_stop	character(len=max_utc_string)	UTC stop time of data sensing
state_vector	character(len=max_utc_string))	UTC of ENVISAT state vector
vector_source	character(len=3)	source of orbit vectors
utc_sbt_time	character(len=max_utc_string)	UTC time corresponding to SBT below
leap_utc	character(len=max_utc_string)	UTC time of the occurrence of the Leap second
cycle	integer(kind=2)	cycle number
leap_sign	integer(kind=2)	Leap second sign
rel_orbit	integer	start relative orbit number
abs_orbit	integer	start absolute orbit number
sat_binary_time	integer	satellite binary time
clock_step	integer	clock step size cock step in picoseconds
tot_size	integer	total size of the product
sph_size	integer	length of the SPH
num_dsd	integer	number of DSD records
dsd_size	integer	length of each DSD record
num_data_sets	integer	number DSs attached
delta_ut	real(kind=8)	DUTl = UTl - UTC
x_position	real(kind=8)	X position in Earth-fixed reference
y_position	real(kind=8)	Y position in Earth-fixed reference
z_position	real(kind=8)	Z position in Earth-fixed reference
x_velocity	real(kind=8)	X velocity in Earth-fixed reference
y_velocity	real(kind=8)	Y velocity in Earth-fixed reference
z_velocity	real(kind=8)	Z velocity in Earth-fixed reference

structure dsd_scia

Name	Type	Description
name	character(len=29)	name of the DSD
type	character(len=2)	type of DSD
flname	character(len=63)	(optional) name of auxiliary file
offset	integer	offset in bytes of the Data Set (DS)
size	integer	size of the DS
num_dsr	integer	number of DS records
dsr_size	integer	size of the DS records (-1 if size is variable)

structure lads_scia

Name	Type	Description
mjd	type(mjd_scia)	Modified Julian data for the year 2000
flag_mds	integer(kind=1)	flag indicating if MDS DSRs are attached
coord	type(coord_scia), dimension(num_coords)	4 corner coordinates

structure mjd_scia

Name	Type	Description
days	integer	number of days elapsed since 1.1.2000 at 00:00 hour
secnd	integer	seconds elapsed since the beginning of the day
musec	integer	microseconds elapsed since the last second

structure coord_scia

Name	Type	Description
lat	integer	longitude (10^{-6} deg)
lon	integer	latitude (10^{-6} deg)

structure Clcon_scia

Name	Type	Description
id	integer(kind=1)	cluster ID (1–64)
channe	integer(kind=1)	channel number (1–8)
type	integer(kind=1)	cluster data type
dummy	integer(kind=1)	dummy to align struct
pixel_	integer(kind=2)	start pixel number
length	integer(kind=2)	cluster length
intg_time	integer(kind=2)	integration time
coaddf	integer(kind=2)	Co-adding factor
n_read	integer(kind=2)	number of cluster readouts per DSR
pet	real(kind=4)	pixel exposure time

structure polV_scia

Name	Type	Description
Q	real(kind=4), dimension(12)	fractional polarisation values Q
error_Q	real(kind=4), dimension(12)	errors on Q values
U	real(kind=4), dimension(12)	fractional polarisation values U
error_U	real(kind=4), dimension(12)	errors on U values
rep_wv	real(kind=4), dimension(13)	representing wavelength for the fractional polarisation values
gdf_para	real(kind=4), dimension(3)	GDF parameters

7.4 Sciamachy Level 1b/c data types

structure sphl_scia

Name	Type	Description
spec_cal	character(len=5)	range of spectral calibration error
saturate	character(len=5)	number of saturated pixel
dark_check	character(len=5)	difference between measured and calibrated dark signal
dead_pixel	character(len=5)	number of dead detector pixels
key_data	character(len=6)	key data version
m_factor	character(len=6)	version of M-factor file
descriptor	character(len=29)	SPH descriptor
start_time	character max_utc_string)	time of the first MDR, in UTC format
stop_time	character max_utc_string)	time of the end of the measurement data in this file
stripline	integer(kind=2)	strip-line counter or zero, if the product is a complete segment
slice_pos	integer(kind=2)	number of the slice, or 1 if no strip-line continuity
no_slice	integer(kind=2)	number of slices
no_nadir	integer(kind=2)	number of nadir measurement states
no_limb	integer(kind=2)	number of limb measurement states
no_occult	integer(kind=2)	number of occultation measurement states
no_monitor	integer(kind=2)	number of monitoring measurement states
no_noproc	integer(kind=2)	number of level 0 MDS, absent in this product
comp_dark	integer(kind=2)	number of processed complete dark states
incomp_dark	integer(kind=2)	number of incomplete dark states
start_lat	real(kind=4)	WGS84 latitude of first nadir point at sensing start time
start_lon	real(kind=4)	WGS84 longitude of first nadir point at sensing stop time
stop_lat	real(kind=4)	WGS84 latitude of last nadir point at sensing stop tim
stop_lon	real(kind=4)	WGS84 longitude of last nadir point at sensing stop time

structure sqads1_scia

Name	Type	Description
mjd	type(mjd_scia)	start time of DSR
flag_mds	integer(kind=1)	flag indicating if MDS DSRs are attached
flag_glint	integer(kind=1)	Sun glint region flag
flag_rainbow	integer(kind=1)	Rainbow region flag
flag_saa_region	integer(kind=1)	SAA region flag
missing_readouts	integer(kind=2)	number of missing readout in state
hotpixel	integer(kind=2), dimension(all_channels)	number of hot pixels per channel and pmd
mean_wv_diff	real(kind=4), dimension(science_channels)	mean value of the wavelength differences
sdev_wv_diff	real(kind=4), dimension(science_channels)	standard deviation of the wavelength differences
mean_diff_leak	real(kind=4), dimension(all_channels)	mean difference of leakage current or offset per channels and pmd

structure ppg_scia

Name	Type	Description
gain_fact	real(kind=4), dimension(science_channels * channel_size)	pixel-to-pixel gain offset
etalon_fact	real(kind=4), dimension(science_channels * channel_size)	etalon correction factor
etalon_resid	real(kind=4), dimension(science_channels * channel_size)	etalon residual
wls_deg_fact	real(kind=4), dimension(science_channels * channel_size)	WLS degradation factor
bad_pixel	integer(kind=1), dimension(science_channels * channel_size)	bad pixel mask

structure srs_scia

Name	Type	Description
sun_spec_id	character(len=3)	Sun spectral identifier
avg_azim_pos	real(kind=4)	average azimuth mirror position
avg_elev_pos	real(kind=4)	average elevation mirror position
avg_solar_ele_ang	real(kind=4)	average Solar elevation angle
dopp_shift	real(kind=4)	Doppler shift at 500 nm
wvlen_sun	real(kind=4), dimension (science_channels * channel_size)	wavelength of the Sun measurement
mean_sun	real(kind=4), dimension (science_channels * channel_size)	mean Sun reference spectrum
precision_sun	real(kind=4), dimension (science_channels * channel_size)	radiometric precision of the mean Sun reference spectrum
accuracy_sun	real(kind=4), dimension (science_channels * channel_size)	radiometric accuracy of the mean Sun reference spectrum
etalon	real(kind=4), dimension (science_channels * channel_size)	diffuser/small aperture etalon
pmd_mean	real(kind=4), dimension(pmd_number)	mean value of the corresponding OMD measurements
pmd_out_nd_out	real(kind=4), dimension(pmd_number)	pmd out-of-band signals with ND ou
pmd_out_nd_in	real(kind=4), dimension(pmd_number)	pmd out-of-band signals with ND in

structure sfp_scia

Name	Type	Description
pix_pos_slit_fun	integer(kind=2)	pixel position for which the slit function is given
type_slit_fun	integer(kind=1)	type of slit function
fwhm_slit_fun	real(kind=4)	FWHM of the slit function (pixel)
f_voi_fwhm_loren	real(kind=4)	for voigt only : FWHM of Lorenzian part (pixel)

structure state1_scia

Name	Type	Description
mjd	type(mjd_scia)	start time of DSR
Clcon	type(Clcon_scia), dimension(max_cluster)	cluster configuration
flag_mds	integer(kind=1)	flag indicating if MDS DSRs are attached
flag_reason	integer(kind=1)	reason code if the attachment flag is set to '1'
type_mds	integer(kind=1)	MDS for this state (nadir, limb, ...)
dummy	integer(kind=1)	dummy to align struct
category	integer(kind=2)	measurement category
state_id	integer(kind=2)	state ID
dur_scan	integer(kind=2)	duration of the scan phase of the state
longest_intg_time	integer(kind=2)	longest integration time
num_clus	integer(kind=2)	number of clusters
num_aux	integer(kind=2)	number of auxiliary data sets
num_pmd	integer(kind=2)	number of integrated pmd values
num_intg	integer(kind=2)	number of integration times
intg_times	integer(kind=2), dimension(max_cluster)	integration times (reverse order)
num_polar	integer(kind=2), dimension(max_cluster)	number of fractional polarization times
total_polar	integer(kind=2)	number of fractional polarization times
num_dsr	integer(kind=2)	number of DSRs
indx	integer	index of this state (starting from zero)
length_dsr	integer	length of this DSR in bytes
is_level_1c	integer	index of this state (starting from zero)
offset	integer(kind=4)	offset to MDS records of this state in bytes
offs_pmd	integer(kind=4)	offset to pmd MDS records of this state in bytes (level 1c only)
offs_polV	integer(kind=4)	offset to polV MDS records of this state in bytes (level 1c only)
orbit_phase	real(kind=4)	orbit phase after eclipse of the state

structure cal_options

Name	Type	Description
start_time	type(mjd_scia)	filter start time
stop_time	type(mjd_scia)	filter stop time
l1b_prod_name	character(len=63)	level 1b product name
geo_filter	integer(kind=1)	geolocation filter flag
time_filter	integer(kind=1)	time filter flag
category_filter	integer(kind=1)	category filter flag
nadir_mds	integer(kind=1)	process Nadir MDS flag
limb_mds	integer(kind=1)	process Limb MDS flag
occ_mds	integer(kind=1)	process Occultation MDS flag
moni_mds	integer(kind=1)	process Monitor MDS flag
pmd_mds	integer(kind=1)	process pmd MDS flag
frac_pol_mds	integer(kind=1)	process fractional polarisation MDS flag
slit_function	integer(kind=1)	Slit function GADS flag
sun_mean_ref	integer(kind=1)	Sun mean reference GADS flag
leakage_current	integer(kind=1)	Leakage Current calibration flag
spectral_cal	integer(kind=1)	Spectral calibration GADS flag
pol_sens	integer(kind=1)	Polarisation Sensitivity GADS flag
rad_sens	integer(kind=1)	Radiance Sensitivity GADS flag
ppg_etalon	integer(kind=1)	PPG/Etalon GADS flag
mem_effect_cal	integer(kind=1)	Memory effect calibration flag
leakage_cal	integer(kind=1)	Leakage Current calibration flag
straylight_cal	integer(kind=1)	Straylight calibration flag
ppg_cal	integer(kind=1)	PPG/Etalon GADS flag
etalon_cal	integer(kind=1)	Etalon calibration flag
wave_cal	integer(kind=1)	Spectral calibration flag
polarisation_cal	integer(kind=1)	Polarisation correction flag
radiance_cal	integer(kind=1)	Radiance calibration flag
num_nadir	integer(kind=2)	number of Nadir clusters selected
num_limb	integer(kind=2)	number of Limb clusters selected
num_occ	integer(kind=2)	number of Occultation clusters selected
num_moni	integer(kind=2)	number of Monitoring clusters selected
start_lat	integer	start Latitiude
start_lon	integer	start Longitude
end_lat	integer	end Latitude
end_lon	integer	end Longitude
category	integer(kind=2), dimension(5)	selected measurement category
nadir_cluster	integer(kind=1), dimension(max_cluster)	Nadir cluster flags
limb_cluster	integer(kind=1), dimension(max_cluster)	Limb cluster flags
occ_cluster	integer(kind=1), dimension(max_cluster)	Occultation cluster flags
moni_cluster	integer(kind=1), dimension(max_cluster)	Monitoring cluster flags

7.5 Sciamachy Level 1c data types

structure mds1c_scia

Name	Type	Description
mjd	type(mjd_scia)	start time of the DSR
rad_units_flag	integer(kind=1)	pixel units indicator
quality_flag	integer(kind=1)	quality indicator
category	integer(kind=2)	measurement category
state_id	integer(kind=2)	state ID
chan_id	integer(kind=2)	science channel ID
clus_id	integer(kind=2)	cluster ID
num_obs	integer(kind=2)	number of observations
num_pixels	integer(kind=2)	number of pixels
dsr_length	integer	length of this DSR in bytes
orbit_phase	real(kind=4)	orbit phase after eclipse
pixel_ids	integer(kind=2), dimension(:), pointer	pixel IDs for cluster
pixel_wv	real(kind=4), dimension(:), pointer	wavelength for pixels
pixel_wv_err	real(kind=4), dimension(:), pointer	wavelength calibration error
pixel_val	real(kind=4), dimension(:,:), pointer	signal values
pixel_val_err	real(kind=4), dimension(:,:), pointer	signal error values
geoN	type(geoN_scia), dimension(:), pointer	geolocation (Nadir states only)
geoL	type(geoL_scia), dimension(:), pointer	geolocation (Limb and Occultation states only)
geoC	type(geoC_scia), dimension(:), pointer	geolocation (Monitoring states only)

structure geoL_scia

Name	Type	Description
pos_esm	real(kind=4)	position of ESM, compared to zero position
pos_asm	real(kind=4)	position of ASM, compared to zero position
sat_h	real(kind=4)	satellite height at middle of integration time
earth_rad	real(kind=4)	Earth radius at middle of integration time
dopp_shift	real(kind=4)	Doppler shift at 500 nm at middle of integration time
sun_zen_ang	real(kind=4), dimension(3)	solar zenith angles at start/middle/end of integration time
sun_azi_ang	real(kind=4), dimension(3)	solar azimuth angles
los_zen_ang	real(kind=4), dimension(3)	line of sight zenith angles
los_azi_ang	real(kind=4), dimension(3)	line of sight azimuth angles
tan_h	real(kind=4), dimension(3)	tangent height at start/middle/end of ntegration time
sub_sat_point	type(coord_scia)	sub-satellite point at middle of integration time
tang_ground_point	type(coord_scia), dimension(3)	coordinates of the tangent ground point

structure geoN_scia

Name	Type	Description
pos_esm	real(kind=4)	position of ESM, compared to zero position
sat_h	real(kind=4)	satellite height at middle of integration time
earth_rad	real(kind=4)	Earth radius at middle of integration time
sun_zen_ang	real(kind=4), dimension(3)	solar zenith angles at start/middle/end of integration time
sun_azi_ang	real(kind=4), dimension(3)	solar azimuth angles
los_zen_ang	real(kind=4), dimension(3)	line of sight zenith angles
los_azi_ang	real(kind=4), dimension(3)	line of sight azimuth angles
sub_sat_point	type(coord_scia)	sub-satellite point at middle of integration time
corner_coord	type(coord_scia), dimension(4)	corner coordinates
center_coord	type(coord_scia)	center coordinate of the nadir pixel

structure geoC_scia

Name	Type	Description
pos_esm	real(kind=4)	position of ESM, compared to zero position
pos_asm	real(kind=4)	position of ASM, compared to zero position
sun_zen_ang	real(kind=4)	solar azimuth angle at middle of integration time
sub_sat_point	type(coord_scia)	sub-satellite point at middle of integration time

structure mds1c_pmd

Name	Type	Description
mjd	type(mjd_scia)	start time of the DSR
quality_flag	integer(kind=1)	quality indicator
category	integer(kind=2)	measurement category
state_id	integer(kind=2)	state ID
dur_scan	integer(kind=2)	duration of the scan phase
num_pmd	integer(kind=2)	number of integrated pmd values
num_geo	integer(kind=2)	number of geolocation records
dsr_length	integer	length of this DSR in bytes
orbit_phase	real(kind=4)	orbit phase after eclipse
int_pmd	real(kind=4), dimension(:,:,:), pointer	integrated pmd values
geoN	type(geoN_scia), dimension(:), pointer	geolocation (Nadir states only)
geoL	type(geoL_scia), dimension(:), pointer	geolocation (Monitoring states only)

structure mds1c_polV

Name	Type	Description
mjd	type(mjd_scia)	start time of the DSR
quality_flag	integer(kind=1)	quality indicator
category	integer(kind=2)	measurement category
state_id	integer(kind=2)	state ID
dur_scan	integer(kind=2)	duration of the scan phase
num_geo	integer(kind=2)	number of geolocation records
total_polV	integer(kind=2)	number of fractional polarization values per integration time
num_diff_intg	integer(kind=2)	number of different integration times
dsr_length	integer	length of this DSR in bytes
orbit_phase	real(kind=4)	orbit phase after eclipse
intg_times	integer(kind=2), dimension(max_cluster)	integration times
num_polar	integer(kind=2), dimension(max_cluster)	repetition factors
polV	type(polV_scia), dimension(:), pointer	fractional polarization values
geoN	type(geoN_scia), dimension(:), pointer	geolocation (Nadir states only)
geoL	type(geoL_scia), dimension(:), pointer	geolocation (Limb and Occultation states only)

Glossary

defs_scia.f90 definition file of SCIAREADLV1 data fields

Envisat ESA Earth observation satellite, launched on March 1, 2001

Enviview ESA software to view ENVISAT data file content. See http://envisat.esa.int/services/enviview.

ESA European Space Agency (http://www.esa.int)

level 1b Geolocated, timestamped data. See also: http://envisat.esa.int/instruments/sciamachy/data-app/dataprod.html

level 1c Calibrated data. Can be produced from level 1b by applying the ESA Enviview SciaL1c routine or the NL-SCIA-DC processor (http://www.sron.nl/~hees/SciaDC/)

Sciamachy SCanning Imaging Absorptions spectroMeter for Atmospheric CHartographY, instrument onboard Envisat

SciaL1c Software tool to convert level 1b data to level 1c data. Part of Enviview.

PDS Payload Data Segment - Sciamachy data binary file format

References

[1] Bovensmann, H., J. P. Burrows, M. Buchwitz, J. Frerick, S. Noel, V. V. Rozanov, K. V. Chance, and A P. H. Goede, SCIAMACHY: Mission Objectives and Measurement Modes, *J. Atmos. Sci.*, 56, 127-150, 1999

[2] Slijkhuis, S.: Envisat-1 SCIAMACHY Level 0 to 1c Processing Algorithm Theoretical Basis Document, ENV-ATB-DLR-SCIA-0041, issue 2, Deutsches Zentrum für Luft- und Raumfahrt e.V., Oberpfaffenhofen, Germany, 2001. 1773, 1781